#127

AD A 001 711

TECHNICAL REPORT
75-37-OTD

A COMPUTER SIMULATION OF BACTERIAL GROWTH DURING FOOD PROCESSING

by

Edward W. Ross, Jr.

Approved for public release; distribution unlimited.

November 1974

UNITED STATES ARMY
NATICK LABORATORIES
Natick, Massachusetts 01760

U.S. ARMY TROOP SUPPORT COMMAND

Office of the Technical Director

AD/A-001 711

A COMPUTER SIMULATION OF BACTERIAL GROWTH DURING FOOD-PROCESSING

Edward W. Ross, Jr.

Army Natick Laboratories
Natick, Massachusetts

November 1974

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: TR-75-37-CTD
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): "A Computer Simulation of Bacterial Growth During Food-Processing"
5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Edward W. Ross, Jr.
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Office of the Technical Director, U.S. Army Natick Laboratories, Natick, Massachusetts - 01760
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS
12. REPORT DATE
13. NUMBER OF PAGES: 72
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COMPUTER PROGRAMS	BACTERIAL POPULATIONS	TEMPERATURE	BACTERIOLOGY
FOOD PREPARATION	FOODS	TIME	FORTRAN V
FOOD PROCESSING	SAFETY	HYPOTHESES	DEATH (ORGANISM)
SIMULATION	BACTERIA	FEEDING SYSTEMS	SALINITY
GROWTH	POPULATION	FEEDING	DATA ANALYSIS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A COMPUTER SIMULATION OF BACTERIAL GROWTH DURING FOOD–PROCESSING

by

Edward W. Ross, Jr.

Office of the Technical Director
U.S. Army Natick Laboratories
Natick, Massachusetts 01760

Abstract

This report describes a computer program that simulates the growth and decline of bacterial populations in foods during the various operations of food-preparation and processing. By means of this program an existing or proposed food preparation process can be studied, its safety assessed and the critical operations identified. Also hypothetical systems for monitoring the bacterial growth can be tried in the simulation. The basic hypothesis of the simulation is that bacterial growth depends on time and temperature in a food. Detailed instructions for the use of the program are given, and an example is set up and solved.

I. Introduction

In recent years the U. S. Army has undertaken a major effort to improve its food-service system, especially the garrison food-service system which affects the morale of enlisted men so directly. The existing system has been studied extensively and various proposals made for its moderization. Some of the suggested changes in the system are new enough so there is no backlog of civilian experience as a guide to their bacteriological safety. This situation was perceived by members of the Operations Research and Systems Analysis Group and the Food Laboratory at the U. S. Army Natick Laboratories, and a method was sought that would permit rational assessment of the bacteriological safety of these processes. The present work is an attempt by the author to provide such a method.

This report describes a computer program that simulates the growth and decline of bacterial populations in foods during the various operations of food-preparation, i.e., storing, mixing, cooking, cooling, etc. It is useful for at least two purposes. First, it can simulate an existing or proposed food-preparation process and tell whether the process is safe or not. Doing this permits the user to see where-abouts in the process the most dangerous situation occurs and, sometimes, what can be done to remedy it. Second it provides a means for testing the effectiveness of systems for monitoring the bacteriological safety of food-preparation processes.

We must emphasize that in order to simulate a food-preparation process, that process must be very well-defined, much more so, for example, than is done by the recipe for the food. This means that the user may have to gather a lot of information before running a simulation. This is of course inconvenient, but it may have benefits, too. A user of the simulation who is investigating a potentially dangerous situation in preparing a certain food may find that merely defining the process carefully is enough to let him see where the trouble may occur and how serious it is.

The computer program is written in the FORTRAN V language rather than in one of the languages specifically invented for simulation purposes, such as GPSS, SIMSCRIPT, SIMULA or GASP. This choice was dictated partly by the unavailability of some of these compilers for the UNIVAC 1106, (the Natick Laboratories computing system) as well as the author's background, the pressure of time and the nature of the simulation.

The principal hypothesis on which the model is based is that growth (or death) of bacteria in a food during storage depends on time and temperature. This dependence is modelled in the computer program by general formulas containing six constants for each stored food. The values assigned to these constants specify the growth and death behavior of the bacteria in that food. The model can easily handle the situation where the bacterial growth is different in the various foods, provided the differences are ones that can be accurately modelled by changes in the constant values.

At first glance it appears that the principal hypothesis is unduly restrictive in two respects. First, only one organism can be studied. Second, bacterial growth depends on many factors other than time and temperature, such as moisture content, ph, salinity, etc. In fact neither of these apparent difficulties is too serious, for they can usually be avoided by repeated running of the program using different values of the constants for the various foods. For example the growth of two different organisms in a food process can be studied by making two computer runs, assigning different values to the sets of constants that prescribe the growth properties of the two organisms. This procedure will usually be adequate but may fail if the two organisms interact significantly with each other. Again, the effect of two different moisture levels can be studied provided the growth of the organism at different moisture levels can be accurately described by changes in the values of the six constants that define growth behavior. This will often give accurate-enough results but could be misleading if growth is sensitive to moisture and if the moisture in one or more foods changes radically with time.

The general structure of the model is described in Section 2 and a brief outline of the computer program is given in Section 3. Section 4 describes an example of the use of the program and the preparation of the data. The results of this example and some further aspects of the model and program are discussed in Section 5. Appendix I contains the derivation of the principal formulas. In Appendix II there is a detailed description of the input data, keyed to the example described in Section 4. The printout of the results of the example forms Appendix III, and Appendix IV contains a listing of the FORTRAN V program. The set-up of executive control cards to be used when running the program on the Natick Laboratories UNIVAC 1106 System is described in Appendix V.

2. General Structure of the Model

The main general types of entities that occur in the model are batches of food, having one main, and several auxiliary, attributes of interest, and activities performed on these batches of food that cause changes in the attributes. The independent variable, time, also plays an important part. We shall describe the roles of these entities in increasing detail in the following paragraphs.

The independent variable, time, is defined by the discrete (integer-valued) quantity I. The simulation starts at I = 1 and proceeds until I attains a value not exceeding IMAX.

Each batch (packet) of food is assigned an index, J, and has, as its main attribute, bacteria count and, as auxiliary attributes, temperature and weight. The bacteria count is specified by the quantity V(J, I), which is the natural logarithm of the density of organisms (e.g. number of organisms per gram) in the J-th food at the I-th time-step. The temperature of the J-th batch of food at time-step I is Y(J, I), and the weight of the batch is W(J), which is assumed not to depend on the time.

The two principal types of activities that are performed are storage and food-handling. Each storage area is assigned an index, KS, and a temperature, YST(KS), which is the ambient temperature of all food batches in that storage area. Every food is in some storage area while it is in the system. It is important to notice that we regard cooking merely as storage at high temperature. Food-handling activities consist of two operations, namely splitting and blending batches of food. In the splitting operation, a single batch of food is subdivided into several new foods, and in the blending operation several batches of food are mixed together to give one new food-batch. Splitting and blending are each other's mirror images with respect to time and are assumed to occur instantaneously, i.e. in much less than one time-step.

Foods may be in the system at the start of simulation or brought in at a later time, and in either case the initial values of the attributes must be prescribed by the user. Foods may also be created or born by means of splits or blends of foods previously in the system, in which case the values of the attributes are derived from those of the progenitor foods by the formulas described in Appendix I.B. Foods may leave the system by giving birth to other foods at a split or blend, or they may be served, or they may remain in the system (i.e. in some storage) until the end of the simulation.

Bacteria counts may change due to either natural growth (i.e. multiplication) and death or contamination. Natural growth or death is assumed to occur as time passes during storage. The rates of change per time-step, A(I), depends on the present bacteria count and temperature of the food. The temperature of the food is first calculated from its initial temperature, ambient temperature and effective diffusion coefficient. This temperature is then employed in general formulas for A(I), and a simple, approximate, numerical integration of A(I) is used to find the bacteria count, V(J, I). The quantity A(I) describes both the growth rate (when $A(I) > O$) and the death rate (when $A(I) < O$) of the bacteria. The formulas for calculating A(I) contain a number of constants that the user must provide in order to model the growth behavior of the organism being studied. The details of the formulas for calculating natural growth and death of organisms are given in Appendix I.

The other source of change in bacteria counts is the contamination introduced into the food. This may occur gradually in a stored food or instantaneously at a split or blend and is modelled by generating a positive random number and adding it to the value of V. For a food created at a split or blend this is done after the values of V implied by the progenitor foods have been calculated.

In addition to these activities the model also simulates two systems for monitoring the number of organisms. The first is a record-keeping system that keeps track of the lengths of time each food has been exposed to its ambient temperatures, estimates the bacteria count at each handling, compares the estimate with a pre-assigned threshold and gives a signal if the threshold value is exceeded. The second system is intended to model the domestic, or housewife, method. At each handling the threshold bacteria count is compared with the counts of all the foods directly involved in the handling or located in the same storage as any directly – involved food. Again a signal is given if the threshold value is exceeded.

The system makes allowance for errors of various types of means of random numbers, usually obeying a Gaussian (Normal) distribution. The principal kinds of errors in the model are scheduling errors, errors in the attribute values of foods initially in the system or entering it from outside, errors in storage temperatures, random losses of weight (spillage) and increases in contamination during handling (previously mentioned), and variation in sensitivity of the cook in the domestic monitor system.

Finally the model also permits changes in the ambient temperature of each food. Up to five changes to new ambient temperatures at arbitrary times may be made in each food, and random errors may occur in the times of these changes.

3. Brief Sketch of Computer Program

The computer program which carries out the simulation consists of three major parts and two important subroutines. The first major part is the reading and recording of the input data. For each food the input data consists of such information as scheduling, storage area, initial attribute values, constants defining growth rates and data about random errors. Also for each split or blend information is given about which foods are involved in the operation, random errors in its time of occurence and the handling contamination. In addition certain general data such as the number of repetitions of the simulation and the temperatures of the storage areas are also read.

The second main portion consists of deriving necessary scheduling information from the input data and introducing the random errors into that scheduling and into the values of V, Y and W for the foods initially in the system or entering it from outside. For each repetition of the simulation this results in a final schedule and set of input values for the attributes, possibly somewhat different from the schedule and values originally intended. The final schedule and values are printed at the start of each repetition of the simulation.

The third main part of the program is the actual calculation of V, Y and W during both storage and handling. This is largely done by means of the two important subroutines, BACGRO and SPLEND. BACGRO does the entire calculation and printing out of Y, V, A and the ambient temperature at each time-step during the storage of each food. SPLEND carries out the calculations of V, Y and W for the new foods created at a split or blend and incorporates the random-handling contamination and spillage losses into the values of V and W, respectively.

The instructions that simulate the two monitoring activities are spread throughout the program and can be identified by the annotations MONREC for the record-keeping system and MONDOM for the domestic system. The short subroutine SNIFF is part of the domestic system and carries out the determination of whether the cooks sense that a food is "bad" or not.

4. Example

We shall describe a simulation of the preparation of chicken pot pie, following a recipe similar to that in the Army CFPF production guide for pre-cooked, frozen and then reheated chicken pot pie. This is a moderately complicated process and illustrates a number of the features of the program as well as some of the problems that arise in translating a recipe into data that the program can use.

The first step in modelling this process is to study the recipe, list the ingredients and try to get a clear idea of what operations are performed and when they occur in the process. Frequently the recipe is not clear about some of the details of the process, and one must either seek more information or try to make sensible guesses. In the present example we start with a fairly detailed list of ingredients:

60 lbs of chicken

2 lbs of sauce (roux)

23 lbs of stock and seasonings

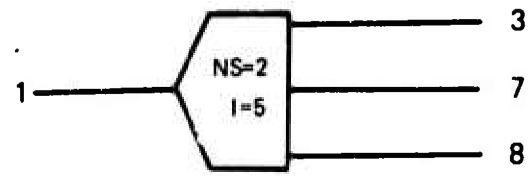
8 lbs of chilled vegetables

6 lbs of frozen peas

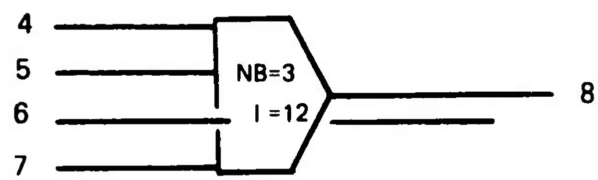
2 lbs of slurry, made of starch and water

5 lbs of topping dough

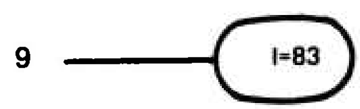
Now it is clear that in listing the ingredients in this way, we have already made some decisions about the amount of detail that we want our model to have. For example we have decided not to bother with the details of how the slurry is made from starch and water and how much seasoning of various kinds goes into the stock. Typically in starting a model a number of such decisions have to be made, in which the gains from added information must be weighed against the cost of greater complexity.



Symbol for a split, occurring at I=5, in which food 1 is split into foods 3, 7 and 8. The split number is NS=2.



Symbol for a blend, occurring at I=12, in which foods 4, 5 and 7 are combined to give food 8. The blend number is NB=3. Notice that food 6 does not take part in the blend.



Symbol for serving of food number 9 at time I=83.

Figure 1: Diagrammatic Symbols for Splits, Blends and Servings

The activities of the recipe can be described as follows: in the central preparation area cook the chicken, remove the chicken from the juices (stock) and let it cool. Combine this stock with the sauce, chilled vegetables and stock and seasoning prepared in advance. Heat the mixture, add frozen peas then heat again and add slurry and the cooked chicken. Heat and add topping, heat again, then allow it to cool. The pies are frozen and transported to the serving area, where they are heated and served.

This description is incomplete since it includes no quantitative information about storage temperatures or times. After further investigation, a description was reached which is best expressed by the diagram shown in Figure 2. This diagram uses the symbols for splits, blends and servings described in Figure 1. In this diagram the horizontal scale is time-steps of five minutes each.

Thirteen foods in all occur during the preparation, of which five are present at the initial time, namely foods numbered 1 (chicken), 2 (sauce), 3 (stock and seasoning), 5 (chilled vegetables), 7 (frozen peas). Two foods, numbered 9 (slurry) and 12 (topping) are brought in at time steps I = 25 and I = 35, respectively, i.e. about 125 and 175 minutes after the start of the process. Six foods are created during the process and are numbered 4 (extra stock), 10 (de-boned chicken), 13 (final chicken pot pie), 6 (alpha), 8 (beta) and 11 (gamma). The last three are simply mixtures created at intermediate stages during the food process.

Each food is represented by a horizontal line in Figure 2, extending over the time-steps at which that food exists. Each line ends at a split or blend or serving symbol. The storage unit number for that food is written above its line, and the temperatures of the various storage units are shown in the upper right corner of Figure 2. Changes in the ambient temperature are shown by writing the new ambient temperature to the right of a short, vertical, dashed line. For example, we see that food 13, the final chicken pot pie is first cooled at room temperature, since it is in storage unit 3, from time I = 39 to I = 45. It is put in the freezer (which has ambient temperature −18°) from I = 45 to I = 91, then reheated at temperature 175° from I = 91 to 97 and served.

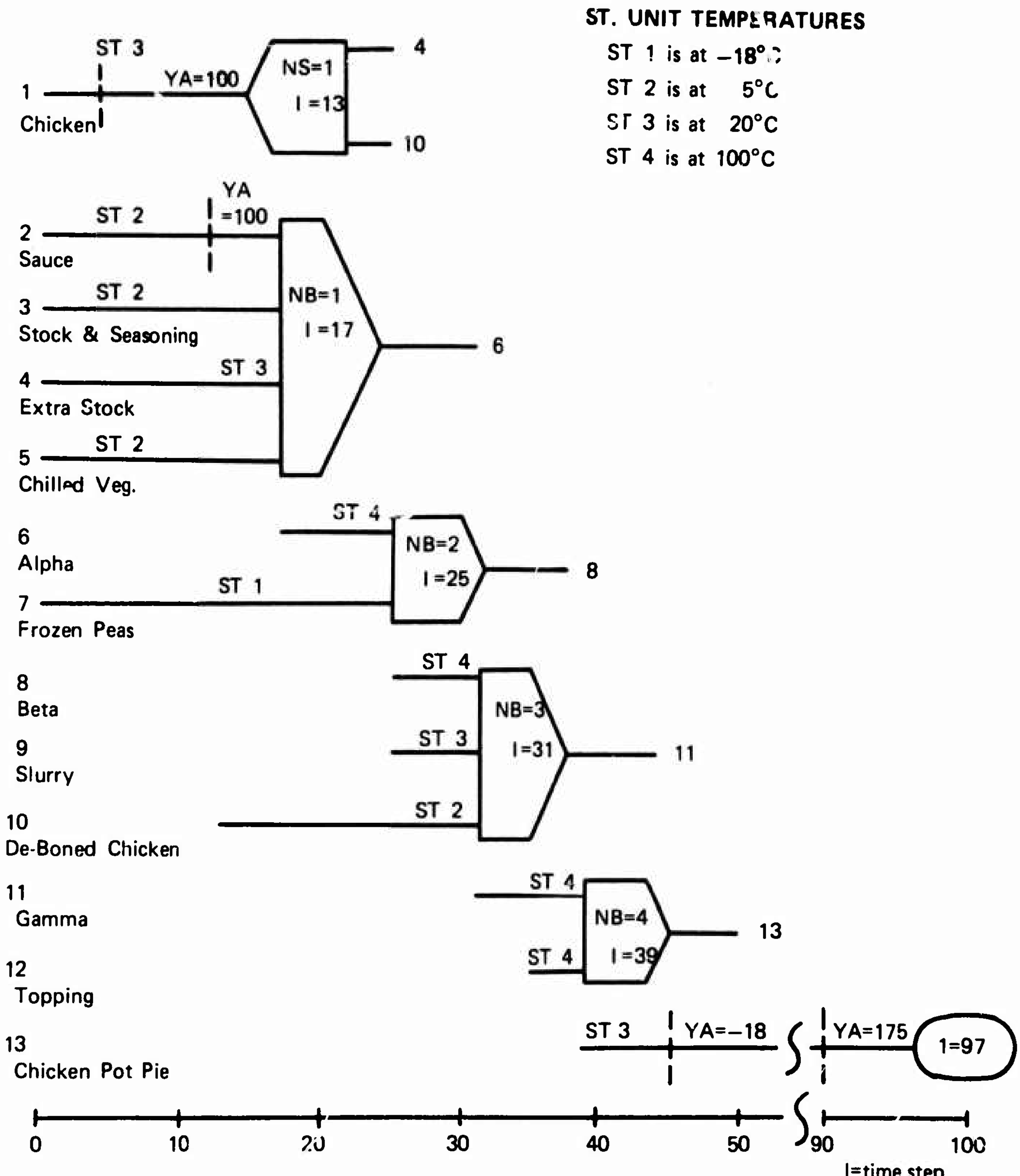


Figure 2: Chicken Pot Pie Preparation Diagram

In the author's experience it is usually very helpful, almost necessary, to prepare a diagram like Figure 2 if the process being studied is at all complicated. With such a diagram it is fairly straightforward to prepare the input data although some additional information, such as the initial attribute values, has to be added to that of the diagram. In the present example the initial temperature, Y, bacteria count, V, and weight, W, are listed in Table I for each food that is not created from existing foods during the process.

Appendix II contains a complete description of the input data for this simulation. Five repetitions of the entire simulation are run, with different random numbers at each repetition. The output from one of the repetitions is listed in Appendix III. Graphs of Y and V as functions of time for certain of the foods are shown in Figure 3.

5. Discussion

The principal results of the chicken pot pie simulation are listed in Appendix III, arising from the input data of Appendix II. Some of the numbers used in this example, (particularly those connected with the action of the monitor systems) are not realistic because they were chosen to demonstrate certain aspects of the program. For example, the constants P(J, K), describing the growth of the organisms, are very roughly those of E. coli at five-minute time-steps. It is possible that we would not regard food as being truly spoiled if it contained 1200 organisms per gram, which is what we mean by choosing TRT = 7.150. The choice TRT = 7.150 was made to ensure that we obtained some cases that we could regard as truly spoiled.

Examination of the results of the simulation show that the final food, number 13, chicken pot pie, has very low bacteria counts, much less than one per gram. The highest bacteria counts were always obtained in the chilled vegetables, food number 5, mostly because these were brought into the system with a high count (V = 7.0 initially). In a sense, therefore, this example is a rather uninteresting one because the food that is served is very safe.

A possible use of this simulation is to investigate how "stable" this process is by seeing how much the results change when we alter some of the input data. For example, we might greatly reduce the final reheating by setting the last storage temperature for food number 13 at 50°C instead of 175°C. We might also permit larger standard deviations

Food Number and name	Y, Initial temp, °C	V, Initial count	W, Weight, lbs
1 chicken	5	6	60
2 Sauce	5	6	2
3 Stock and Seasoning	5	6	23
5 Chilled vegetables	5	7	8
7 Frozen peas	−18	5	6
9 Slurry	10	6	2
12 Topping	5	6	5

Table 1: Initial Attributes of Foods

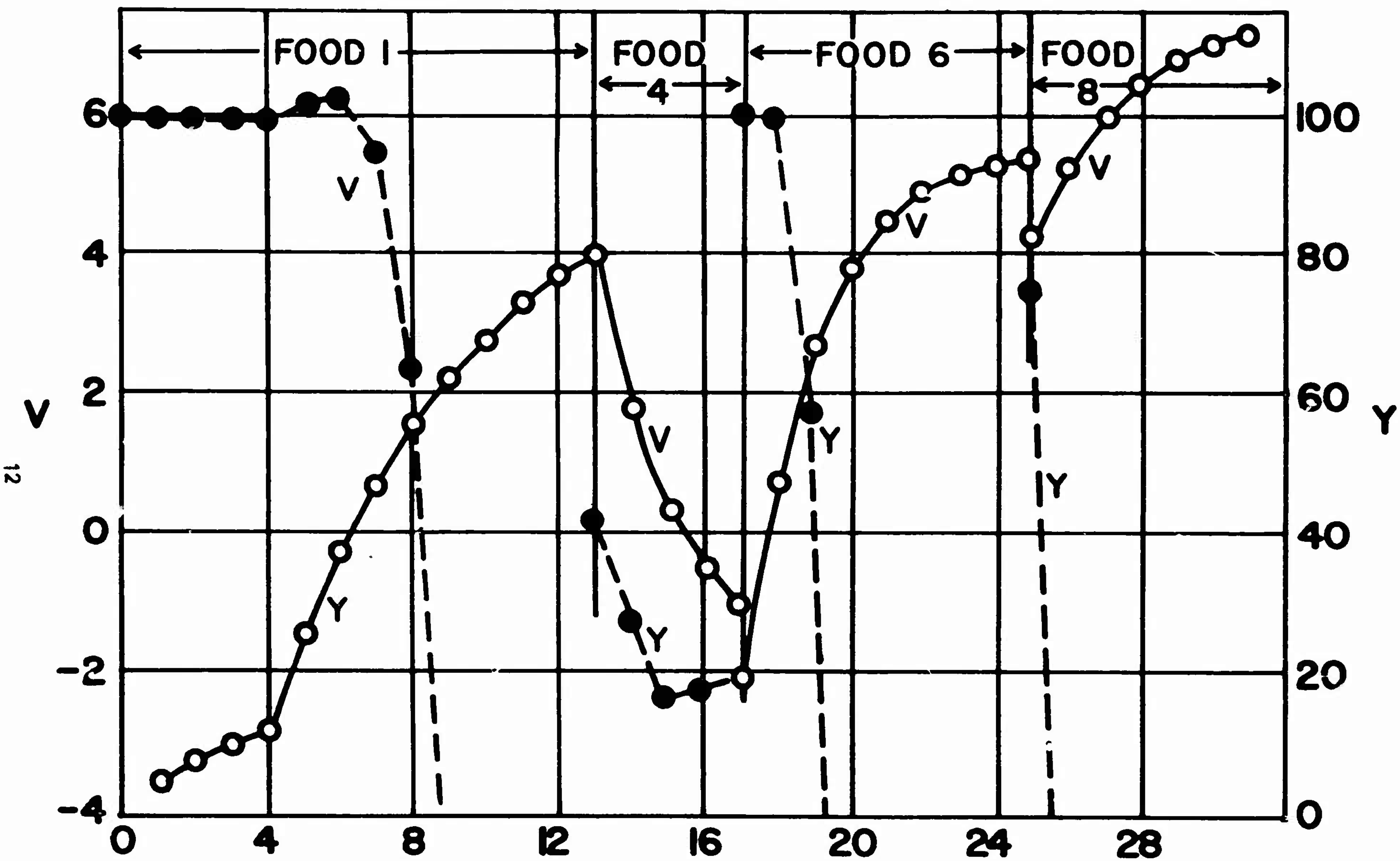


Figure 3: Graphs of Y (Temperature in °C) and V (Natural Logarithm of Bacteria Count) versus I (time step) in Foods 1, 4, 6 and 8

in some of the random numbers, like those governing the storage temperatures and the handling errors. In this way we could see how large the bacteria counts become as we let the process become more sloppy.

It is perhaps worth emphasizing that the two monitor-systems give signals when the bacteria count is high enough, regardless of how close the food is to being served. Food that is highly contaminated and about to be served is certainly more dangerous than an equally contaminated food that will later be cooked before it is served. However, the two monitoring systems ignore that distinction. It would not be too difficult to change the monitor systems so they only operated on foods that are served or so they became increasingly sensitive as the process went on.

It is evident that the program as described here is workable. Undoubtedly improvements or extensions of it are possible, but it is necessary to try it first on a great variety of processes to see what changes need to be made. The author hopes that it will be convenient and useful in its present form.

Appendix I: Formulas Used in the Simulation

In this Section we give more complete descriptions of the calculation of bacterial change during storage and at a handling operation. Also we describe the two monitoring systems more thoroughly than before.

A. Bacterial Growth During Storage of a Food

We assume that the differential equation governing the non-random growth of organism population is

$$d\rho/dt = \rho\alpha[Y(t)]$$

where ρ = density of organisms (i.e. number of organisms per gram), t = time, and $\alpha[Y(t)]$ is a function that depends on the temperature, Y, which itself depends on t. This equation can be written

$$dV/dt = \alpha[Y(t)]$$

if we define

$$V = \log_e \rho.$$

Integrating this, we obtain

$$V(t) = V(t_p) + \int_{s=t_p}^{t} \alpha[Y(s)]\,ds,$$

where t_p is any time earlier than t. We now change to a discrete approximation in which time is only defined at t_1, t_2, t_3, ... where for all i

$$t_i - t_{i-1} = \Delta t.$$

If we let $t = t_i$ and $t_p = t_{i-1}$, abbreviate $V(t_i)$ by V_i and $V(t_{i-1})$ by V_{1-1} and use the trapezoidal approximation for the integral, we obtain

$$V_i = V_{i-1} + ½ \Delta t \quad \alpha[Y(t_i)] + \alpha[Y(t_{i-1})]$$

Then, defining for all i

$$A_i = \Delta t \ \alpha[Y(t_i)],$$

we get

$$V_i = V_{i-1} + ½ \ (A_i + A_{i-1}).$$

We add to this the random change, r_i, to obtain the formula

$$V_i = V_{i-1} + ½ \ (A_i + A_{i-1}) + r_i,$$

where r_i are uncorrelated Gaussian random variables with mean m_r, standard deviation σ_r.

We now write down the specific formulas defining A_i. The temperature of a food is found from

$$Y_i = Y_s + (Y_1 - Y_s)e^{-P_1(i-i_1)}$$

where Y_s is the ambient temperature in the storage area, i is the present time-step, i_1 is the time-step at which the food entered the storage area, Y_1 is the temperature of the food at time i_1 and P_1 is the effective thermal diffusion coefficient of the food. A_i is expressed in terms of Y_i by means of

$$A_i = 0 \quad \text{if} \quad Y_i < P_2$$

$$A_i = P_3 \exp -P_4(P_5-Y_i)^2 \ \text{if} \ P_2 \leqslant Y_i \leqslant P_5$$

$$A_i = P_3-P_6(Y_i-P_5)^3 \qquad \text{if} \ Y_i \nearrow P_5$$

In these formulas P_2 is the temperature below which (it is assumed) there is negligible growth or death. P_5 is the temperature at maximum growth, and P_3 is the maximum value of A_i, i.e. the value of A_i when $Y_i = P_5$. P_4 determines how rapidly A_i decreases from P_3 when Y_i decreases from P_5. P_6 determines how quickly A_i decreases from P_3 as Y_i increases above P_5. When

$$Y_i > P_5 + (P_3/P_6)^{1/3},$$

A_i becomes negative, and $|A_i|$ is proportional to death rate rather than growth rate.

The following relationship is sometimes useful in estimating the value of A. Suppose that a food is held at a constant temperature such that $A > 0$, long enough so that the number of bacteria is just double its initial value, i.e. V has increased by $\ln 2 = .693$. If n time-steps are required to accomplish this, then the (constant) value of A is

$$A = (\ln 2)/n = .693/n$$

and $n\Delta t$ is the generation time, usually called g_2. Similarly, if $A < 0$ and n is the number of time steps needed to reduce the bacteria count to half its initial value, then

$$A = -.693/n$$

and $n\Delta t$ is the half-life of the organism population.

B. Attributes of Quantities Created During Handling

Here we record the formulas used by the subroutine SPLEND in deriving the temperature, Y, weight, W, and bacteria count, V, of new foods created at a split or blend.

At a split a single food, which exists and has attribute values Y_O, W_O, V_O, disappears, and N new foods are created with attribute values Y_K, W_K, V_K where K = 1, 2 ... N. We define

$$f_K = W_K/W_O,$$

i.e. f_K are the proportions by weight of the new foods, and

$$f_1 + f_2 + \dots + f_N = 1.$$

These proportions are specified by the user. Then aside from handling errors the attributes of the new foods are taken to be

$$Y_K = Y_O$$

$$V_K' = V_O \qquad K = 1, 2, \dots N$$

$$W_K' = f_K W_O$$

Handling errors are assumed to occur in V_K and W_K and to have the form

$$V_K = V_K' + |r_{VK}|$$

$$W_K = W_K' \; (1-|r_{WK}|)$$

where r_{VK} and r_{WK} are Gaussian – distributed random numbers with zero mean but different variances. The nature of these errors follows from the assumption that handling the food can result only in increased counts (i.e. handling contamination) and decreased weight (spillage).

At a blend N foods with attributes Y_K, W_K and V_K, K = 1, 2, ... N, disappear, and a single food with attribute values Y_O, and W_O and V_O is created. Disregarding handling errors temporarily, we calculate W_O, Y_O and V_O as follows. The weight of the new food is the sum of the weights of the foods being blended, i.e.

$$W_O' = W_1 + W_2 + \dots + W_N = \Sigma W_K$$

The temperature of the new food is the weighted average of the temperatures of the foods entering the blend, i.e.

$$Y_O = \frac{W_1 Y_1 + W_2 Y_2 + \dots + W_N Y_N}{W_1 + W_2 + \dots + W_N} = \frac{\Sigma W_K Y_K}{W_O'}$$

To calculate V_O, we recall that the density of bacteria in the K-th food is ρ_K,

$$\rho_K = e^{V_K},$$

and the total number of bacteria in the K-th food, ν_K, is the density multiplied by the weight,

$$\nu_K = W_K \rho_K = W_K e^{V_K}.$$

The total number of bacteria in the new food, ν_O, is the sum of the numbers in the foods entering the blend,

$$\nu_O = \Sigma \nu_K.$$

Finally V_O is the natural logarithm of the density of organisms in the new food, i.e.

$$V_O' = \log_e (\nu_O / W_O') = \log_e \left[\frac{\Sigma W_K e^{K}}{\Sigma W_K} \right].$$

Handling errors in a blend are assumed to have the same form as in a split, i.e.

$$V_O = V_O' + |r_{VO}|$$

$$W_O = W_O (1-|r_{WO}|)$$

where r_{VO} and r_{WO} are Gaussian – distributed random numbers with zero mean but (in general) different variances.

C. The Two Monitor Systems

The record-keeping monitor system operates as follows: each food is assigned an estimated bacteria count, RI(J), when it first comes into the system. Each food leaving the system before the end of simulation is also assigned a final count RO(J), in a manner to be described shortly. For a food that is in the system at the start of simulation, or is brought in from outside during simulation, the value of RI(J) is assigned by the user, based on previous experience with that type of food. Foods created out of other foods during simulation get their values of RI(J) from the RO(J) values of their progenitor foods in the following ways: if the JO-th food comes from a split in the J-th food

$$RI(JO) = RO(J)$$

If the JU-th food comes from a blend of foods numbered JRI, JRI, ..., JRN, then

$$RI(JU) = \frac{1}{N} [RO(JRI) + ROJR2) + ... + RO(JRN)].$$

We now describe how RO(J) is determined. Three parameters, Q1, Q2, and Q3, are supplied by the user, and a temperature, YCOMP, is taken to be the highest of all the ambient temperatures to which the J-th food is exposed. Then

$$\begin{aligned} RO(J) &= RI(J) \quad \text{if } YCOMP < Q1 \\ &= RI(J) + Q3\ (YCOMP\text{-}Q1)(IT-I1) \\ &\qquad \text{if } Q1 \leq YCOMP \leq Q2 \\ &= 0 \quad \text{if } YCOMP > Q2. \end{aligned}$$

In these formulas Q1 is a temperature below which no growth or death occurs, Q2 is a temperature above which the organism is almost completely killed off, I1 and IT are the time-steps at which the food enters and leaves the system respectively, and Q3 is a coefficient specifying the estimated bacterial growth rate.

In the program the value of RO(J) for each food is compared with the number TESTI, supplied by the user. TESTI is the level of contamination that the monitor system considers to be "bad". If RO(J) exceeds TESTI, this constitutes a signal that the monitor system has detected an unsafe situation in that food. A message to this effect, "MONREC SIGNAL", will appear in the printed output for the food. Also, if several repetitions of the simulation are made, the program prints out NS, the number of repetitions in which this monitor system detected an unsafe situation.

Finally a number called TRT, supplied by the user, is the bacteria count which he takes as the actual count at which a "bad" (dangerous) condition occurs. This need not be the same as TESTI, the number which the monitor system is using for that purpose. The program calculates the largest bacteria count occurring during the simulation and compares it with TRT, to determine whether a dangerous situation actually does occur during the simulation. If several repetitions of the simulation are done, the program prints out NT, the true number of trials in which a dangerous situation occurres. To help in assessing the effectiveness of the monitor system, two other numbers, N11 and N00, are also printed out. N11 is the number of trials in which the monitor system correctly stated that there was a dangerous situation, and N00 is the number of trials in which the monitor system correctly stated that there was no dangerous situation.

The domestic monitor system operates quite differently. It is assumed that at each food-handling, i.e. split or blend, a cook checks the bacteria count of the food coming into the split or emerging from the blend. Conceptually, we imagine that the cook sniffs the food and detects a signal if the food smells "bad". In the program this is done by the subroutine SNIFF, which compares the bacteria count, VT, with a random number, C, that represents the cook's threshold for the detection of bad smells. If VT exceeds C, a detection is made. The threshold, C, is obtained from

$$C = AV + SD*XX$$

where, XX, is a standardized Gaussian random number. Thus, AV represents the average threshold value and SD the standard deviation in the threshold value.

There is a further aspect to the domestic monitor system. It is assumed that at each food handling, the cook goes to all the storage units containing one or more of the foods involved in the split or blend. He then checks all the foods resident in those storage areas at that time. In the program this is done by first determining the storage areas involved, then finding what foods are in these storage areas and finally applying the subroutine SNIFF to these foods.

It is reasonable to expect that the cook's threshold for detection would be higher in the situation just described than in the previous circumstances, where he is checking food that he is actually working on. Accordingly the values of AV and SD are permitted to be different in the two different applications of SNIFF. AV1 and SD1 are the parameters of the threshold random number for foods being directly handled, and AV2 and SD2 are the corresponding parameters when the foods are being only indirectly checked.

When there are several repetitions of the simulation, the program prints out quantities NSD, ND00 and ND11, analogous to the NS, N00 and N11 printed out for the record-keeping monitor system.

Appendix II: Input Data

In discussing the input data format we shall always describe it as if it were stored on cards, because this is the storage medium used in entering data for the first time.

The general arrangement of the input data is as follows. First, there are four cards giving general information about the simulation, such as NJ, the number of separate foods, NMS, the number of splits, NSTOR the number of storage units, as well as information about the two monitor systems. Next are the cards describing the properties of the individual foods. Each food requires at least (usually exactly) six cards to define it. Following these is a single card specifying the temperatures of the storage units. Then come the cards supplying information about the splits, three cards per split, and finally the cards giving the data for the blends, again three cards for each blend.

There are only two essentially different formats for input cards. Most of the cards are in free-field format, i.e. they consist of numbers (some with decimal points, some without) separated by commas, and without any blanks. All the remaining cards are of "Legend" type, that is, they are messages, made up of ordinary words, used to annotate the printed output of the simulations.

It is necessary to be careful about the order in which the individual foods are entered. The program assigns J=1 as the number of the first food entered, J=2 that of second food and so on. The main body of the program consists of calculations done in the following order:

(a) BACGRO for J=1 (i.e. bacterial growth of food number 1 during storage)

(b) SPLEND for J=1 (i.e. split or blend whose input is J=1, if it can be carried out)

(c) BACGRO for J=2

(d) SPLEND for J=2

etc.

Because of this arrangement the foods must be entered in such an order that foods created at splits or blends are entered after their progenitor foods. For example, if cake flour is created by a split in flour, and the data for cake flour were inadvertently entered before flour, then the program would assign J=1 to cake flour and J=2 to flour. The subroutine SPLEND would then wrongly derive the initial bacterial count and temperature for flour from those of cake flour.

This requirement, that foods created at a split or blend be entered after their progenitor foods, is not a severe one and tends to agree with what a user would ordinarily do in any case. The main advantage of preparing a diagram like Figure 2 is that one can easily check that this condition is satisfied by making sure, that the numbers of the foods on the right of (emerging from) each split or blend are higher than those on the left (entering). When the diagram is such that this condition is satisfied at all the splits and blends, then entering the foods from top to bottom of the diagram will give a correct order.

Incidentally, an alert reader will notice that in Figure 2 all the foods entering a blend are numbered consecutively. This condition makes the diagram clear and easy to read, but the program does not require it. I.e., the program will function correctly even though the foods entering a blend are not numbered consecutively, provided the previous condition is met.

We now describe the information on each card, and give as an example the corresponding information for the chicken pot pie simulation of Section 4. The cards are grouped according to the general arrangement stated near the beginning of this Appendix. Following this detailed description of the input is a printed listing of all the input data for the simulation of chicken pot pie.

Group A: General Information

Card 1: Legend Card. Any desired message can be punched in columns 2–72 and will be printed at the top of the first output page.

Example: CHICKEN POT PIE SIMULATION
is punched in columns 2–27

Card 2: NJ, NMS, NMB, X(1), IMAX, SDV, NREP, NRNO, IPTRD, NSTOR

Example: 13, 1, 4, 347., 100, .05, 5, 130, 0, 4
is punched in columns 1–29

Here

NJ = number of separate foods = 13

NMS = number of splits =1

NMB = number of blends = 4

X(1) = the "seed" for random-number generation = 347.

IMAX = an integer larger than the total number of time-steps in the simulation = 100

SDV = standard deviation of the random error in the bacterial growth equation = .05

NREP = number of repetitions of the simulation = 5

NRNO = number of random numbers generated at each repetition = 130

IPTRD = controller for printing. If IPTRD = 1, printing of Y, V and YA is suppressed except at the initial and final times for each food. Choosing IPTRD=0 in the example permits printing of all values of Y, V and YA.

NSTOR= number of storage units = 4

Card 3: AV1, SD1, AV2, SD2

Example: 12., 1., 13., 1
is punched in columns 1–13

See the discussion of domestic monitor system. Appendix I.C.

AV1	=	average threshold count for directly-handled foods = 12.
SD1	=	standard deviation in these threshold counts = 1.
AV2	=	average threshold count for foods not directly handled = 13.
SD2	=	standard deviation in these threshold counts = 1.

Card 4: Q1, Q2, Q3, TESTI, TRT

Example: 10., 50., .01, 12., 12.
is punched in columns 1–19

See the discussion of the record-keeping monitor system, Appendix I.C.

Q1	=	lower end of temperature growth range = 10.
Q2	=	upper end of temperature growth range = 50.
Q3	=	bacterial growth rate = .01
TESTI	=	threshold of dangerous contamination used by monitor = 12.
TRT	=	true threshold of dangerous contamination = 12.

NOTE 1: The simulation assumes that NMS $\geq$ 1 and NMB $\geq$ 1, see Group A, Card 2, i.e. there is a least one split and one blend. These must be introduced if they do not occur naturally. For example, if one wants to simply find the growth of bacteria in chicken, then one must introduce a one-for-one split and one-for-one blend, giving the chicken three different food numbers in the simulation. This means that we must always have NJ $\geq$ 3.

Group B: Information specific to a Food.

For each food the following set of cards must be prepared. The examples given are for the first food in the chicken pot pie simulation.

Card 1: Legend Card. Any desired description or message can be punched in columns 2–72 and will be printed along with the value of J at several points in the output.

Example: CHICKEN
is punched in colums 9–15

Card 2: IC(J), W(J), Y(J,1), V(J,1), ISER(J), MST(J)

Example: 1, 60., 5., 6., 100, 3
is punched in columns 1–17

IC(J) = the time step at which food is first in the system = 1
W(J) = the intended weight of the food = 60.
Y(J,1) = the temperature at the initial time step = 5.
V(J,1) = the initial bacteria count = 6.
ISER(J) = is an integer which is the time-step at which the food is served. If the food is not served (as in this example) ISER(J) must be at least as large as IMAX, ISER(J) = 100
MST(J) = the storage unit where this food is stored = 3

Card 3: ICD(J), WD(J), YD(J), VD(J), NATC(J), CONT(J)

Example: 0, 0., 0., 0., 1, 0
is punched in columns 1–15

ICD(J) = standard deviation of random error in initial time = 0.
WD(J) = standard deviation of random error in weight = 0.
YD(J) = standard deviation of random error in initial temperature = 0
VD(J) = standard deviation of random error in initial bacteria count = 0.
NATC(J) = the number of ambient temperature changes in the food = 1
CONT(J) = the mean of the random increase in bacteria count = 0.

Card 4: P(J,1), P(J,2), P(J,3), P(J,4), P(J,5), P(J,6)

Example: .17, 0., .15, .0047, 40., .000443
is punched in columns 1–28

For the J-th food these are the constants referred to as $P_1, P_2, \ldots P_6$ in Appendix I.A. They describe approximately the behavior of E. coli at five-minute time steps.

P(J,1)	=	thermal diffusion coefficient = .17
P(J,2)	=	temperature below which no growth or death occurs = 0.
P(J,3)	=	maximum value of growth rate = .15
P(J,4)	=	coefficient of growth rate change below maximum growth temperature = .0047
P(J,5)	=	temperature of maximum growth rate = 40.
P(J,6)	=	coefficient of growth rate change above maximum growth temperature = .000443

Card 5: RI(J)

Example: 6
is punched in columns 1–2

RI(J) = initial bacteria count assumed by record-keeping monitor = 6.

Card 6: IAC(J,1), IACD(J,1), YAC(J,1)

Example: 4, 0, 100
is punched in columns 1–8

IAC(J,1)	=	time step at which first ambient temperature change occurs = 4
IACD(J,1)	=	standard deviation of the random error in the time step at which first ambient temperature change occurs = 0
YAC(J,1)	=	the ambient temperature after its first change = 100.

Note 1: If more than one ambient temperature change occurs, one additional card, exactly like card 6, should be made up for each additional temperature change and added following card 6. The cards describing ambient temperature changes should be in order of increasing values of IAC.

Note 2: If no ambient temperature change occurs, card 6 should contain the entries 0, 0, 0. in columns 1–6. Card 6 is then the final card for that food.

Note 3: For foods not created during the simulation the program uses all the information on these input cards. For foods created during the simulation the quantities W(J), Y(J,1) and V(J,1) on card 2 of this group are not used. Instead the program obtains these quantities from the attributes of the progenitor foods by using subroutine SPLEND. Hence any reasonable values can be entered for these quantities, in particular 0., 0., 0.

Group C: Information about Storage Units

This group contains a single card with 2*NSTOR numbers, giving the intended ambient temperature and standard deviation of the random error in that temperature for each of the NSTOR storage units.

Card 1: YST(1), YSTD(1), YST(2), YSTD(2), ... , YST(NSTOR), YSTD(NSTOR)

Example: −18., 0., 5., 0., 20., 5., 100., 10
is punched in columns 1–29

YST(1) = ambient temperature in storage unit 1 = −18.
YSTD(1) = standard deviation of random error in that temperatu = 0.
YST (2) = ambient temperature in storage unit 3 = 5.
YSTD(2) = standard deviation of random error in that temperature = 0.
etc.

Group D: Information about Splits

The split number, MS, is assigned to each split in order as its data is read in. Exactly three cards define the data of each split. See Appendix I.B.

Card 1: NJO(MS)

Example: 2
is punched in column 1

NJO(MS) = the number of different foods created at split number MS = 2

Card 2: IS(MS), JSI(MS), ISD(MS), JSO(MS,1), PR(MS,1), JSO(MS,2), PR(MS,2), ...

Example: 13, 1, 0, 4, .3, 10, .6

IS(MS)	=	time-step at which split number MS occurs = 13
JSI(MS)	=	number of the food being split = 1
ISD(MS)	=	standard deviation of the random error in the time of the split = 0
JSO(MS,1)	=	number of the first food created at the split = 4
PR(MS,1)	=	proportion (by weight) of the first food created at the split = .3
JSO(MS,2)	=	number of the second food created at the split = 10
PR(MS,2)	=	proportions of the second food created = .6

Card 3: HEVS(MS,1), HEWS(MS,1), HEVS(MS,2), HEWS(MS,2) ...

Example: .3, 0.0, .5, .05

HEVS(MS,1)	=	standard deviation of the random handling contamination in the first food created at the split = .3
HEWS(MS,1)	=	standard deviation of the random spillage weight loss in the first food created at the split = 0.0
HEVS(MS,2)	=	standard deviation of contamination in the second food created at the split = .5
HEWS(MS,2)	=	standard deviation of spillage loss in the second food created at the split = .05

NOTE: The time of the split, IS(MS), must always agree with the initial times, IC(J), of all the foods created at the split. Also ISD(MS) must agree with ICD(J) for all the foods created at the split, and the latter must agree with each other.

Group E: Information about Blends

The blend number is assigned to each blend in order as its data is read in. Exactly three cards define the data of each blend. Sec Appendix I.B.

Card 1: NJI(MB)

Example: 4
is punched in column 1

NJI(MB) = the number of different foods blended together at blend MB = 4

Card 2: IB(MB), JBO(MB), IBD(MB), JBI(MB,1), JBI(MB,2), JBI(MB,3), JBI(MB,4)

Example: 17, 6, 0, 2, 3, 4, 5

IB(MB) = time step at which blend number MB occurs = 17
JBO(MB) = the number of the food created at the blend = 6
IBD(MB) = standard deviation of the random error in the time of the blend = 0
JBI(MB,1) = number of the first food blended = 2
JBI(MB,2) = number of the second food blended = 3
etc.

Card 3: HEVB(MB), HEWB(MB)

Example: .4, .02

HEVB(MB) = standard deviation of the random handling contamination in the food created at blend MB = .4
HEWB(MB) = standard deviation of the random spillage weight loss in the food created at the blend = .02

NOTE: The time of the blend, IB(MB), must always agree with the initial time, IC(J), of the food created at the blend. Also IBD(MB), its standard deviation, must agree with ICD(J) for the food created at the blend.

This concludes the description of the input. Below is a complete listing of the input data for the chicken pot pie simulation. Each line of printing represents the data punched on one input card.

```
 CHICKEN POT PIE SIMULATION
13,1,4,347.,100.,05,5,130,0,4
6.05,0.,7.2,0.
10.,50.,.01,5.,7.15
          CHICKEN
1,60.,5.,6.,100,3
0,0.,0.,0.,1,0.
.17,0.,.15,.0047,40.,.000443
6.
4,0,100.
          SAUCE (ROUX)
1,2.,5.,6.,100,2
0,0.,0.,0.,1,0.
.50,0.,.15,.0047,40.,.000443
6.
13,1,100.
          STOCK AND SEASONING
1,23.,5.,6.,100,2
0,0.,0.,0.,0,0.
.50,0.,.15,.0047,40.,.000443
6.
0,0,0.
          EXTRA STOCK
13,0.,0.,6.,100,3
1,0.,0.,0.,0,0.
.50,0.,.15,.0047,40.,.000443
6.
0,0,0.
          CHILLED VEGETABLES
1,8.,5.,7.,100,2
0,0.,0.,0.,0,0.
.50,0.,.15,.0047,40.,.000443
6.
0,0,0.
          MIXTURE ALPHA
17,0.,0.,0.,100,4
1,0.,0.,0.,0,0.
.50,0.,.15,.0047,40.,.000443
6.
0,0,0.
          FROZEN PEAS
1,6.,-18.,5.,100,1
0,0.,0.,0.,0,0.
.50,0.,.15,.0047,40.,.000443
6.
0,0,0.
          MIXTURE BETA
25,0.,0.,0.,100,4
1,0.,0.,0.,0,0.
.40,0.,.15,.0047,40.,.000443
6.
0,0,0.
          SLURRY
25,2.,10.,6.,100,3
```

```
0,0.,0.,0.,1,0.
.40,0.,.15,.0047,40.,.000443
6.
27,0,100.
        DE-BONED, COOKED CHICKEN
13,0.,0.,0.,100,2
1,0.,0.,0.,1,0.
.17,0.,.15,.0047,40.,.000443
6.
27,0,20.
        MIXTURE GAMMA
31,0.,0.,0.,100,4
1,0.,0.,0.,0,0.
.30,0.,.15,.0047,40.,.000443
6.
0,0,0.
        TOPPING
35,5.,5.,6.,100,4
0,0.,0.,0.,0,0.
.17,0.,.15,.0047,40.,.000443
6.
0,0,0.
        CHICKEN POT PIE
39,0.,0.,0.,97,3
2,0.,0.,0.,2,0.
.20,0.,.15,.0047,40.,.000443
6.
45,2,-18.
91,2,175.
-18.,0.,5.,0.,20.,5.,100.,10.
2
13,1,0,4,.3,10,.6
.3,0.0,.5,.05
4
17,6,0,2,3,4,5
.4,.02
2
25,8,0,6,7
.3,.01
3
31,11,0,8,9,10
.3,.05
2
39,13,0,11,12
.4,.04
```

Appendix III: Output

In this Appendix we discuss the printed output from the simulation of chicken pot pie described in Section 4. The general form of the output is sketched in the following paragraphs, and the actual print-out from a small simulation is shown. The Appendix concludes with some detailed comments about this output.

The output can be divided into two main sections. The first contains a listing of all the information that the user furnishes about the simulation, i.e. a methodical record of the input data. Within this section there are a number of subsections, conforming with the grouping of input data in Appendix II. The first subsection contains the descriptive title of the run, followed by four lines of general information. The data pertaining to the individual foods is listed next, consisting of an annotative heading plus (usually) five lines of data about each food. The third subsection presents the data on the storage units, and the last subsection, headed "Intended Schedule of Splits and Blends", summarizes the information about splits and blends.

The second main section describes the actual (as contrasted with intended) events of the simulation and its results. It begins with the table headed "Final Schedule", followed by a summary of information about each food after the introduction of random errors. The next subsection contains the calculated time-dependence of Y, food temperature, V, logarithmic bacteria count, A, growth rate and YA, storage temperature, for each food during its storage, followed by what happens to the food at the end of storage. This subsection contains the principal results of the simulation. Each repetition of the simulation leads to another section of output starting with the "Final Schedule" table. After the output from all repetitions is finished, there is a final, two-line, subsection summarizing the effectiveness of the two monitor systems.

At this point it is useful for the reader to examine the output for the sample run of the chicken pot pie simulation whose input was listed at the end of Appendix 2. The reader will notice in the line after the heading that NREP=5, i.e. this run included 5 repetitions of the simulation. In this report only the printout from the first and fourth repetitions are shown in order to save space.

```
@ADD,PE MIMAGE.CHIPOTPIE
 CHICKEN POT PIE SIMULATION
 NJ=13 NMS= 1 NMB= 4 X(1)=    347. IMAX= 100 SDV=  .050000 NREP=    5
 NRNO=130 IPTRD=0 NSTOR= 4
 AV1=   6.050 SD1=     .000 AV2=   7.200 SD2=     .000
 Q1= 10.0000 Q2= 50.0000 Q3=   .0100 TEST1=  5.0000 TRT=  7.1500

 J= 1           CHICKEN
 IC=  1 W=  60.0000 YI=  5.0000 VI= 6.0000 ISER=100 STOR. NO.= 3
 ICD= 0 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 1 CONT=  .0000
 P(J,K)    .17000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  4 IACD= 0 YAC= 100.0000
 RI=    6.0000

 J= 2           SAUCE (ROUX)
 IC=  1 W=   2.0000 YI=  5.0000 VI= 6.0000 ISER=100 STOR. NO.= 2
 ICD= 0 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 1 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC= 13 IACD= 1 YAC= 100.0000
 RI=    6.0000

 J= 3           STOCK AND SEASONING
 IC=  1 W=  23.0000 YI=  5.0000 VI= 6.0000 ISER=100 STOR. NO.= 2
 ICD= 0 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 0 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  0 IACD= 0 YAC=    .0000
 RI=    6.0000

 J= 4           EXTRA STOCK
 IC= 13 W=    .0000 YI=   .0000 VI= 6.0000 ISER=100 STOR. NO.= 3
 ICD= 1 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 0 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  0 IACD= 0 YAC=    .0000

 J= 5           CHILLED VEGETABLES
 IC=  1 W=   8.0000 YI=  5.0000 VI= 7.0000 ISER=100 STOR. NO.= 2
 ICD= 0 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 0 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  0 IACD= 0 YAC=    .0000
 RI=    6.0000

 J= 6           MIXTURE ALPHA
 IC= 17 W=    .0000 YI=   .0000 VI=   .0000 ISER=100 STOR. NO.= 4
 ICD= 1 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 0 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  0 IACD= 0 YAC=    .0000

 J= 7           FROZEN PEAS
 IC=  1 W=   6.0000 YI=-18.0000 VI= 5.0000 ISER=100 STOR. NO.= 1
 ICD= 0 WD=   .0000 YD=   .0000 VD=   .0000 NATC= 0 CONT=  .0000
 P(J,K)    .50000    .00000    .15000    .00470  40.00000    .00044
 K= 1 IAC=  0 IACD= 0 YAC=    .0000
 RI=    6.0000
```

```
J= 8             MIXTURE BETA
IC= 25 W=     .0000 YI=    .0000 VI=  .0000 ISER=100 STOR. NO.= 4
ICD= 1 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 0 CONT=   .0000
P(J,K)     .40000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC=  0 IACD= 0 YAC=     .0000

J= 9             SLURRY
IC= 25 W=   2.0000 YI= 10.0000 VI= 6.0000 ISER=100 STOR. NO.= 3
ICD= 0 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 1 CONT=   .0000
P(J,K)     .40000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC= 27 IACD= 0 YAC= 100.0000

J=10             DE-BONED, COOKED CHICKEN
IC= 13 W=     .0000 YI=    .0000 VI=  .0000 ISER=100 STOR. NO.= 2
ICD= 1 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 1 CONT=   .0000
P(J,K)     .17000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC= 27 IACD= 0 YAC=  20.0000

J=11             MIXTURE GAMMA
IC= 31 W=     .0000 YI=    .0000 VI=  .0000 ISER=100 STOR. NO.= 4
ICD= 1 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 0 CONT=   .0000
P(J,K)     .30000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC=  0 IACD= 0 YAC=     .0000

J=12             TOPPING
IC= 35 W=   5.0000 YI=  5.0000 VI= 6.0000 ISER=100 STOR. NO.= 4
ICD= 0 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 0 CONT=   .0000
P(J,K)     .17000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC=  0 IACD= 0 YAC=     .0000

J=13             CHICKEN POT PIE
IC= 39 W=     .0000 YI=    .0000 VI=  .0000 ISER= 97 STOR. NO.= 3
ICD= 2 WD=    .0000 YD=    .0000 VD=   .0000 NATC= 2 CONT=   .0000
P(J,K)     .20000      .00000      .15000      .00470  40.00000     .00044
K= 1 IAC= 45 IACD= 2 YAC= -18.0000
K= 2 IAC= 91 IACD= 2 YAC= 175.0000

STORAGE UNIT  1 IS AT TEMP -18.000 WITH STD DEV   .0000

STORAGE UNIT  2 IS AT TEMP   5.000 WITH STD DEV   .0000

STORAGE UNIT  3 IS AT TEMP  20.000 WITH STD DEV  5.0000

STORAGE UNIT  4 IS AT TEMP 100.000 WITH STD DEV 10.0000

     INTENDED SCHEDULE OF SPLITS AND BLENDS

SPLIT NUMBER 1 IN FOOD NUMBER  1 OCCURS AT TIME  13 WITH STD DEV   0
THE NUMBERS AND PROPORTIONS OF THE RESULTING FOODS ARE
  FOOD NUMBER  4   PROPORTION  .30000
  FOOD NUMBER 10   PROPORTION  .60000
HANDLING CONTAMINATION    .30000,  SPILLAGE WEIGHT LOSS     .00000
HANDLING CONTAMINATION    .50000,  SPILLAGE WEIGHT LOSS     .05000

BLEND 1 PRODUCES FOOD  6 AT TIME  17 WITH S.D.   0 FROM FOODS
  2   3   4   5
HANDLING CONTAMINATION    .40000,  SPILLAGE WEIGHT LOSS     .02000
```

```
BLEND 2 PRODUCES FOOD  8 AT TIME  25 WITH S.D.   0 FROM FOODS
  6   7
HANDLING CONTAMINATION     .30000,  SPILLAGE WEIGHT LOSS      .01000

BLEND 3 PRODUCES FOOD 11 AT TIME  31 WITH S.D.   0 FROM FOODS
  8   9  10
HANDLING CONTAMINATION     .30000,  SPILLAGE WEIGHT LOSS      .05000

BLEND 4 PRODUCES FOOD 13 AT TIME  39 WITH S.D.   0 FROM FOODS
 11  12
HANDLING CONTAMINATION     .40000,  SPILLAGE WEIGHT LOSS      .04000

     FINAL SCHEDULE
FOOD  1 STARTS AT I=  1 AND ENDS AT SPLIT  1 STOR. NO.= 3
FOOD  2 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  3 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  4 STARTS AT I= 13 AND ENDS AT BLEND  1 STOR. NO.= 3
FOOD  5 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  6 STARTS AT I= 17 AND ENDS AT BLEND  2 STOR. NO.= 4
FOOD  7 STARTS AT I=  1 AND ENDS AT BLEND  2 STOR. NO.= 1
FOOD  8 STARTS AT I= 25 AND ENDS AT BLEND  3 STOR. NO.= 4
FOOD  9 STARTS AT I= 25 AND ENDS AT BLEND  3 STOR. NO.= 3
FOOD 10 STARTS AT I= 13 AND ENDS AT BLEND  3 STOR. NO.= 2
FOOD 11 STARTS AT I= 31 AND ENDS AT BLEND  4 STOR. NO.= 4
FOOD 12 STARTS AT I= 35 AND ENDS AT BLEND  4 STOR. NO.= 4
FOOD 13 STARTS AT I= 39 AND IS SERVED AT I= 97 STOR. NO.= 3

J= 1          CHICKEN
IC=  1 IE= 13 LS= 1 LB= 0 NATC= 1
W=  60.0000 YI=   5.0000 VI=   6.0000 YA=  22.3770
K= 1 IACH=  4 YAC=100.00000

J= 2          SAUCE (ROUX)
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 1
W=   2.0000 YI=   5.0000 VI=   6.0000 YA=   5.0000
K= 1 IACH= 13 YAC=100.00000

J= 3          STOCK AND SEASONING
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 0
W=  23.0000 YI=   5.0000 VI=   6.0000 YA=   5.0000

J= 4          EXTRA STOCK
IC= 13 IE= 17 LS= 0 LB= 1 NATC= 0
W=    .0000 YI=    .0000 VI=    .0000 YA=  21.9099

J= 5          CHILLED VEGETABLES
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 0
W=   8.0000 YI=   5.0000 VI=   7.0000 YA=   5.0000

J= 6          MIXTURE ALPHA
IC= 17 IE= 25 LS= 0 LB= 2 NATC= 0
W=    .0000 YI=    .0000 VI=    .0000 YA=  95.8263

J= 7          FROZEN PEAS
IC=  1 IE= 25 LS= 0 LB= 2 NATC= 0
W=   6.0000 YI= -18.0000 VI=   5.0000 YA= -18.0000
```

```
J= 8           MIXTURE BETA
IC= 25 IE= 31 LS= 0 LB= 3 NATC= 0
W=     .0000 YI=     .0000 VI=     .0000 YA= 114.0200

J= 9           SLURRY
IC= 25 IE= 31 LS= 0 LB= 3 NATC= 1
W=    2.0000 YI=  10.0000 VI=    6.0000 YA=  27.1119
K= 1 IACH= 27 YAC=100.00000

J=10           DE-BONED, COOKED CHICKEN
IC= 13 IE= 31 LS= 0 LB= 3 NATC= 1
W=     .0000 YI=     .0000 VI=     .0000 YA=   5.0000
K= 1 IACH= 27 YAC= 20.00000

J=11           MIXTURE GAMMA
IC= 31 IE= 39 LS= 0 LB= 4 NATC= 0
W=     .0000 YI=     .0000 VI=     .0000 YA= 103.5712

J=12           TOPPING
IC= 35 IE= 39 LS= 0 LB= 4 NATC= 0
W=    5.0000 YI=    5.0000 VI=    6.0000 YA= 104.1283

J=13           CHICKEN POT PIE
IC= 39 IE= 97 LS= 0 LB= 0 NATC= 2
W=     .0000 YI=     .0000 VI=     .0000 YA=  19.2408
K= 1 IACH= 46 YAC=-18.00000
K= 2 IACH= 88 YAC=175.00000
```

FOOD NUMBER 1 CHICKEN

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	22.37698	1
2	7.71663	5.97024	.00112	22.37698	2
3	10.00856	5.92052	.00219	22.37698	3
4	11.94218	5.92616	.00371	22.37698	4
5	25.70872	5.95663	.05744	100.00000	5
6	37.32306	6.15091	.14503	100.00000	6
7	47.12167	6.22938	-.01001	100.00000	7
8	55.38841	5.47188	-1.46430	100.00000	8
9	62.36277	2.33034	-4.80428	100.00000	9
10	68.24680	-4.96316	-9.83416	100.00000	10
11	73.21094	-17.87334	-16.07734	100.00000	11
12	77.39901	-37.35074	-23.02310	100.00000	12
13	80.93234	-64.02174	-30.23110	100.00000	13

```
RI=    6.0000 RO=     .0000 VE=  -64.0217

THIS FOOD ENDS IN SPLIT  1 AT TIME, I=  13. INPUT TO SPLIT HAS
  TEMP=Y=  80.932   BAC. COUNT=V= -64.022 WEIGHT=W=  60.000
K AND LIND(K) 1   0     2   1     3   1     4   0
OUTPUT FROM SPLIT IS AS FOLLOWS
  FOOD NO.  4, TIME= 13, Y=  80.932, V=    .216, W=  18.000
  FOOD NO. 10, TIME= 13, Y=  80.932, V=    .289, W=  35.918
```

FOOD NUMBER 2 SAUCE (ROUX)

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	5.00000	1
2	5.00000	6.00949	.00047	5.00000	2
3	5.00000	6.04303	.00047	5.00000	3

4	5.00000	6.01281	.00047	5.00000	4
5	5.00000	6.13485	.00047	5.00000	5
6	5.00000	6.12334	.00047	5.00000	6
7	5.00000	6.19452	.00047	5.00000	7
8	5.00000	6.23755	.00047	5.00000	8
9	5.00000	6.24146	.00047	5.00000	9
10	5.00000	6.30309	.00047	5.00000	10
11	5.00000	6.34361	.00047	5.00000	11
12	5.00000	6.28498	.00047	5.00000	12
13	5.00000	6.23829	.00047	5.00000	13
14	42.37959	6.34732	.14403	100.00000	14
15	65.05145	3.00684	-6.81470	100.00000	15
16	78.80264	-13.11792	-25.73138	100.00000	16
17	87.14315	-49.13859	-46.26512	100.00000	17

RI= 6.0000 RO= .0000 VE= -49.1386

FOOD NUMBER 3 STOCK AND SEASONING

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	5.00000	1
2	5.00000	5.99103	.00047	5.00000	2
3	5.00000	6.02379	.00047	5.00000	3
4	5.00000	6.05223	.00047	5.00000	4
5	5.00000	6.06751	.00047	5.00000	5
6	5.00000	6.14797	.00047	5.00000	6
7	5.00000	6.16938	.00047	5.00000	7
8	5.00000	6.15180	.00047	5.00000	8
9	5.00000	6.10543	.00047	5.00000	9
10	5.00000	6.07439	.00047	5.00000	10
11	5.00000	6.04348	.00047	5.00000	11
12	5.00000	6.12607	.00047	5.00000	12
13	5.00000	6.18875	.00047	5.00000	13
14	5.00000	6.17484	.00047	5.00000	14
15	5.00000	6.24848	.00047	5.00000	15
16	5.00000	6.23094	.00047	5.00000	16
17	5.00000	6.16659	.00047	5.00000	17

RI= 6.0000 RO= 6.0000 VE= 6.1666
MONREC SIGNAL

FOOD NUMBER 4 EXTRA STOCK

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	80.93234	.21616	-.59223	21.90990	13
2	57.70882	-1.29140	-2.31021	21.90990	14
3	43.62304	-2.33449	.12893	21.90990	15
4	35.07958	-2.21477	.13387	21.90990	16
5	29.89771	-2.03550	.09285	21.90990	17

RI= .0000 RO= .4764 VE= -2.0355

FOOD NUMBER 5 CHILLED VEGETABLES

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	7.00000	.00047	5.00000	1
2	5.00000	7.06141	.00047	5.00000	2
3	5.00000	7.07714	.00047	5.00000	3
4	5.00000	6.97111	.00047	5.00000	4
5	5.00000	7.00327	.00047	5.00000	5
6	5.00000	6.94106	.00047	5.00000	6
7	5.00000	7.02143	.00047	5.00000	7
8	5.00000	6.91424	.00047	5.00000	8

9	5.00000	7.01987	.00047	5.00000	9
10	5.00000	7.08657	.00047	5.00000	10
11	5.00000	7.10434	.00047	5.00000	11
12	5.00000	7.09268	.00047	5.00000	12
13	5.00000	7.07536	.00047	5.00000	13
14	5.00000	7.13735	.00047	5.00000	14
15	5.00000	7.08740	.00047	5.00000	15
16	5.00000	7.12174	.00047	5.00000	16
17	5.00000	7.12076	.00047	5.00000	17

RI= 6.0000 RO= 6.0000 VE= 7.1208
MONREC SIGNAL

THIS FOOD ENDS IN BLEND NO. 1. INPUT TO BLEND HAS
FOOD NO. 2, TIME= 17, Y= 87.143, V= -49.139, WT.= 2.000
FOOD NO. 3, TIME= 17, Y= 5.000, V= 6.167, WT.= 23.000
FOOD NO. 4, TIME= 17, Y= 29.898, V= -2.035, WT.= 18.000
FOOD NO. 5, TIME= 17, Y= 5.000, V= 7.121, WT.= 8.000
K AND LIND(K) 1 0 2 1 3 1 4 1
OUTPUT FROM BLEND HAS
FOOD NO.= 6, TIME= 17, Y= 17.009, V= 6.024, WT.= 49.344

FOOD NUMBER 6 MIXTURE ALPHA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	17.00873	6.02421	.01251	95.82632	17
2	48.02103	6.00319	-.07861	95.82632	18
3	66.83095	1.75707	-8.40681	95.82632	19
4	78.23974	-14.74459	-24.62128	95.82632	20
5	85.15952	-47.39148	-40.64919	95.82632	21
6	89.35658	-94.23379	-53.11461	95.82632	22
7	91.90222	-151.66308	-61.78862	95.82632	23
8	93.44623	-216.30290	-67.48243	95.82632	24
9	94.38272	-285.61029	-71.10028	95.82632	25

RI= 3.1191 RO= .0000 VE= -285.6103

FOOD NUMBER 7 FROZEN PEAS

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	-18.00000	5.00000	.00000	-18.00000	1
2	-18.00000	4.97947	.00000	-18.00000	2
3	-18.00000	4.84784	.00000	-18.00000	3
4	-18.00000	4.89009	.00000	-18.00000	4
5	-18.00000	4.98503	.00000	-18.00000	5
6	-18.00000	5.00463	.00000	-18.00000	6
7	-18.00000	4.91015	.00000	-18.00000	7
8	-18.00000	4.95256	.00000	-18.00000	8
9	-18.00000	4.93969	.00000	-18.00000	9
10	-18.00000	4.95704	.00000	-18.00000	10
11	-18.00000	5.00335	.00000	-18.00000	11
12	-18.00000	5.06126	.00000	-18.00000	12
13	-18.00000	5.01780	.00000	-18.00000	13
14	-18.00000	5.03225	.00000	-18.00000	14
15	-18.00000	5.02824	.00000	-18.00000	15
16	-18.00000	5.11189	.00000	-18.00000	16
17	-18.00000	5.13433	.00000	-18.00000	17
18	-18.00000	5.11170	.00000	-18.00000	18
19	-18.00000	5.08629	.00000	-18.00000	19
20	-18.00000	5.12079	.00000	-18.00000	20
21	-18.00000	5.12691	.00000	-18.00000	21

22	-18.00000	5.10189	.00000	-18.00000	22
23	-18.00000	5.14211	.00000	-18.00000	23
24	-18.00000	5.18777	.00000	-18.00000	24
25	-18.00000	5.15849	.00000	-18.00000	25

RI= 6.0000 RO= 6.0000 VE= 5.1585
MONREC SIGNAL

THIS FOOD ENDS IN BLEND NO. 2. INPUT TO BLEND HAS
FOOD NO. 6, TIME= 25, Y= 94.383, V=-285.610, WT.= 49.344
FOOD NO. 7, TIME= 25, Y= -18.000, V= 5.158, WT.= 6.000
K AND LIND(K) 1 1 2 0 3 0 4 1
OUTPUT FROM BLEND HAS
FOOD NO.= 8, TIME= 25, Y= 82.199, V= 3.433, WT.= 55.092

FOOD NUMBER 8 MIXTURE BETA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	82.19894	3.43263	-.63887	114.01999	25
2	92.68970	-29.18136	-64.65088	114.01999	26
3	99.72187	-108.65348	-94.21347	114.01999	27
4	104.43567	-215.01946	-118.36759	114.01999	28
5	107.59543	-342.60241	-136.67193	114.01999	29
6	109.71348	-485.96942	-149.94077	114.01999	30
7	111.13325	-640.52156	-159.29894	114.01999	31

RI= 3.0000 RO= .0000 VE= -640.5216

FOOD NUMBER 9 SLURRY

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	10.00000	6.00000	.00218	27.11190	25
2	15.64145	6.00523	.00923	27.11190	26
3	19.42303	6.00461	.02050	27.11190	27
4	45.98764	6.12632	.05490	100.00000	28
5	63.79443	3.26576	-5.81801	100.00000	29
6	75.73068	-9.72106	-20.05820	100.00000	30
7	83.73179	-38.23226	-36.90062	100.00000	31

RI= 6.0000 RO= .0000 VE= -38.2323

FOOD NUMBER 10 DE-BONED, COOKED CHICKEN

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	60.93234	.28885	-.59223	5.00000	13
2	69.06144	-5.33824	-10.72315	5.00000	14
3	59.04639	-12.16957	-2.91085	5.00000	15
4	50.59703	-13.85776	-.37718	5.00000	16
5	43.46861	-13.95789	.13151	5.00000	17
6	37.45462	-13.89031	.14550	5.00000	18
7	32.38082	-13.81857	.11418	5.00000	19
8	28.10023	-13.76476	.07710	5.00000	20
9	24.48885	-13.69590	.04842	5.00000	21
10	21.44206	-13.60814	.02972	5.00000	22
11	18.87159	-13.63677	.01840	5.00000	23
12	16.70297	-13.59368	.01170	5.00000	24
13	14.87338	-13.51779	.00772	5.00000	25
14	13.32983	-13.50370	.00530	5.00000	26
15	12.02758	-13.50436	.00379	5.00000	27
16	13.27395	-13.49092	.00523	20.00000	28
17	14.32547	-13.48179	.00677	20.00000	29
18	15.21260	-13.44995	.00836	20.00000	30
19	15.96104	-13.49983	.00992	20.00000	31

RI= .0000 RO= 1.8000 VE= -13.4998

THIS FOOD ENDS IN BLEND NO. 3. INPUT TO BLEND HAS
FOOD NO. 8, TIME= 31, Y= 111.133, V=-640.522, WT.= 55.092
FOOD NO. 9, TIME= 31, Y= 83.732, V= -38.232, WT.= 2.000
FOOD NO. 10, TIME= 31, Y= 15.961, V= -13.500, WT.= 35.918
K AND LIND(K) 1 0 2 1 3 1 4 1
OUTPUT FROM BLEND HAS
FOOD NO.=11, TIME= 31, Y= 73.791, V= .191, WT.= 89.711

FOOD NUMBER 11 MIXTURE GAMMA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	73.79101	.19133	-.35583	103.57116	31
2	81.50949	-15.74546	-31.53442	103.57116	32
3	87.22747	-54.76420	-46.51463	103.57116	33
4	91.46346	-108.11862	-60.23105	103.57116	34
5	94.60156	-174.10405	-71.96387	103.57116	35
6	96.92632	-250.90991	-81.57274	103.57116	36
7	98.64854	-336.32293	-89.21661	103.57116	37
8	99.92439	-428.51825	-95.17672	103.57116	38
9	100.86957	-525.97297	-99.75896	103.57116	39

RI= .6000 RO= .0000 VE= -525.9730

FOOD NUMBER 12 TOPPING

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	104.12827	35
2	20.49724	6.04972	.02510	104.12827	36
3	33.57171	6.11109	.12352	104.12827	37
4	44.60218	6.22913	.10682	104.12827	38
5	53.90820	5.72607	-1.04184	104.12827	39

RI= 6.0000 RO= .0000 VE= 5.7261

THIS FOOD ENDS IN BLEND NO. 4. INPUT TO BLEND HAS
FOOD NO. 11, TIME= 39, Y= 100.870, V=-525.973, WT.= 89.711
FOOD NO. 12, TIME= 39, Y= 53.908, V= 5.726, WT.= 5.000
K AND LIND(K) 1 0 2 0 3 1 4 1
OUTPUT FROM BLEND HAS
FOOD NO.=13, TIME= 39, Y= 98.390, V= 3.307, WT.= 88.701

FOOD NUMBER 13 CHICKEN POT PIE

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	98.39039	3.30671	-1.36038	19.24077	39
2	84.04300	-16.23512	-37.69725	19.24077	40
3	72.29635	-42.55360	-14.77327	19.24077	41
4	62.67900	-52.41697	-5.01744	19.24077	42
5	54.80499	-55.55599	-1.28757	19.24077	43
6	48.35829	-56.26345	-.10868	19.24077	44
7	43.08018	-56.23576	.13705	19.24077	45
8	38.75883	-56.01516	.14892	19.24077	46
9	28.47020	-55.85092	.08031	-18.00000	47
10	20.04658	-55.79982	.02309	-18.00000	48
11	13.14990	-55.81018	.00506	-18.00000	49
12	7.50338	-55.82317	.00105	-18.00000	50
13	2.88041	-55.77293	.00023	-18.00000	51
14	-.90457	-55.79520	.00000	-18.00000	52
15	-4.00345	-55.77630	.00000	-18.00000	53
16	-6.54059	-55.75830	.00000	-18.00000	54

17	-8.61783	-55.75044	.00000	-18.00000	55
18	-10.31853	-55.68500	.00000	-18.00000	56
19	-11.71094	-55.63998	.00000	-18.00000	57
20	-12.85096	-55.69514	.00000	-18.00000	58
21	-13.78432	-55.68975	.00000	-18.00000	59
22	-14.54849	-55.59563	.00000	-18.00000	60
23	-15.17414	-55.54278	.00000	-18.00000	61
24	-15.68639	-55.61711	.00000	-18.00000	62
25	-16.10577	-55.77279	.00000	-18.00000	63
26	-16.44914	-55.74161	.00000	-18.00000	64
27	-16.73026	-55.79869	.00000	-18.00000	65
28	-16.96043	-55.69548	.00000	-18.00000	66
29	-17.14887	-55.61345	.00000	-18.00000	67
30	-17.30315	-55.58629	.00000	-18.00000	68
31	-17.42947	-55.62767	.00000	-18.00000	69
32	-17.53289	-55.70799	.00000	-18.00000	70
33	-17.61756	-55.66743	.00000	-18.00000	71
34	-17.68689	-55.69729	.00000	-18.00000	72
35	-17.74364	-55.69900	.00000	-18.00000	73
36	-17.79011	-55.78619	.00000	-18.00000	74
37	-17.82816	-55.67530	.00000	-18.00000	75
38	-17.85931	-55.62170	.00000	-18.00000	76
39	-17.88481	-55.58746	.00000	-18.00000	77
40	-17.90569	-55.58092	.00000	-18.00000	78
41	-17.92279	-55.48587	.00000	-18.00000	79
42	-17.93678	-55.52533	.00000	-18.00000	80
43	-17.94824	-55.57060	.00000	-18.00000	81
44	-17.95762	-55.48525	.00000	-18.00000	82
45	-17.96531	-55.47280	.00000	-18.00000	83
46	-17.97160	-55.45072	.00000	-18.00000	84
47	-17.97674	-55.48998	.00000	-18.00000	85
48	-17.98096	-55.56344	.00000	-18.00000	86
49	-17.98441	-55.47215	.00000	-18.00000	87
50	-17.98724	-55.43489	.00000	-18.00000	88
51	16.99541	-55.56133	.01247	175.00000	89
52	45.63679	-55.54091	.07066	175.00000	90
53	69.08636	-60.86979	-10.75114	175.00000	91
54	88.28525	-91.17761	-49.72088	175.00000	92
55	104.00396	-173.92798	-116.00137	175.00000	93
56	116.87336	-332.54395	-201.09789	175.00000	94
57	127.40993	-580.95062	-295.70990	175.00000	95
58	136.03655	-924.92157	-392.23589	175.00000	96
59	143.09942	-1363.70110	-485.33123	175.00000	97

THIS FOOD IS SERVED AT TIME 97
RI= .0000 RO= .0000 VE=-1363.7011
MAX. COUNT IS 7.1373 IN FOOD NO. 5 AT TIME 14
NSIG=1 NSIGD=0 NTRU=0

```
     FINAL SCHEDULE
FOOD  1 STARTS AT I=  1 AND ENDS AT SPLIT  1 STOR. NO.= 3
FOOD  2 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  3 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  4 STARTS AT I= 13 AND ENDS AT BLEND  1 STOR. NO.= 3
FOOD  5 STARTS AT I=  1 AND ENDS AT BLEND  1 STOR. NO.= 2
FOOD  6 STARTS AT I= 17 AND ENDS AT BLEND  2 STOR. NO.= 4
FOOD  7 STARTS AT I=  1 AND ENDS AT BLEND  2 STOR. NO.= 1
FOOD  8 STARTS AT I= 25 AND ENDS AT BLEND  3 STOR. NO.= 4
FOOD  9 STARTS AT I= 25 AND ENDS AT BLEND  3 STOR. NO.= 3
FOOD 10 STARTS AT I= 13 AND ENDS AT BLEND  3 STOR. NO.= 2
FOOD 11 STARTS AT I= 31 AND ENDS AT BLEND  4 STOR. NO.= 4
FOOD 12 STARTS AT I= 35 AND ENDS AT BLEND  4 STOR. NO.= 4
FOOD 13 STARTS AT I= 39 AND IS SERVED AT I= 97 STOR. NO.= 3

J= 1              CHICKEN
IC=  1 IE= 13 LS= 1 LB= 0 NATC= 1
W=  60.0000 YI=   5.0000 VI=   6.0000 YA=  21.6280
K= 1 IACH=  4 YAC=100.00000

J= 2              SAUCE (ROUX)
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 1
W=   2.0000 YI=   5.0000 VI=   6.0000 YA=   5.0000
K= 1 IACH= 13 YAC=100.00000

J= 3              STOCK AND SEASONING
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 0
W=  23.0000 YI=   5.0000 VI=   6.0000 YA=   5.0000

J= 4              EXTRA STOCK
IC= 13 IE= 17 LS= 0 LB= 1 NATC= 0
W=  18.0000 YI=  80.2978 VI=    .1111 YA=  21.9506

J= 5              CHILLED VEGETABLES
IC=  1 IE= 17 LS= 0 LB= 1 NATC= 0
W=   8.0000 YI=   5.0000 VI=   7.0000 YA=   5.0000

J= 6              MIXTURE ALPHA
IC= 17 IE= 25 LS= 0 LB= 2 NATC= 0
W=  49.8154 YI=  13.7372 VI=   5.9348 YA=  98.7461

J= 7              FROZEN PEAS
IC=  1 IE= 25 LS= 0 LB= 2 NATC= 0
W=   6.0000 YI= -18.0000 VI=   5.0000 YA= -18.0000

J= 8              MIXTURE BETA
IC= 25 IE= 31 LS= 0 LB= 3 NATC= 0
W=  55.1513 YI= 100.5085 VI=   2.9985 YA= 102.2474

J= 9              SLURRY
IC= 25 IE= 31 LS= 0 LB= 3 NATC= 1
W=   2.0000 YI=  10.0000 VI=   6.0000 YA=  21.6481
K= 1 IACH= 27 YAC=100.00000

J=10              DE-BONED, COOKED CHICKEN
```

IC= 13 IE= 31 LS= 0 LB= 3 NATC= 1
W= 33.4602 YI= 11.9689 VI= .3913 YA= 5.0000
K= 1 IACH= 27 YAC= 20.00000

J=11 MIXTURE GAMMA
IC= 31 IE= 39 LS= 0 LB= 4 NATC= 0
W= 88.5702 YI= 71.9381 VI= .0524 YA= 100.6015

J=12 TOPPING
IC= 35 IE= 39 LS= 0 LB= 4 NATC= 0
W= 5.0000 YI= 5.0000 VI= 6.0000 YA= 97.8588

J=13 CHICKEN POT PIE
IC= 39 IE= 97 LS= 0 LB= 0 NATC= 2
W= 89.6401 YI= -17.9916 VI= 3.4269 YA= 23.8291
K= 1 IACH= 45 YAC=-18.00000
K= 2 IACH= 93 YAC=175.00000

FOOD NUMBER 1 CHICKEN

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	21.62800	1
2	7.59954	6.00614	.00108	21.62800	2
3	9.79268	5.93523	.00206	21.62800	3
4	11.64296	5.97603	.00343	21.62800	4
5	25.45627	6.01859	.05551	100.00000	5
6	37.11008	6.15310	.14423	100.00000	6
7	46.94199	6.29487	.00180	100.00000	7
8	55.23682	5.54775	-1.41706	100.00000	8
9	62.23488	2.48154	-4.71977	100.00000	9
10	68.13890	-4.70745	-9.72018	100.00000	10
11	73.11991	-17.55583	-15.94427	100.00000	11
12	77.32221	-36.99811	-22.88064	100.00000	12
13	80.86755	-63.42109	-30.08706	100.00000	13

RI= 6.0000 RO= .0000 VE= -63.4211

THIS FOOD ENDS IN SPLIT 1 AT TIME, I= 13. INPUT TO SPLIT HAS
TEMP=Y= 80.868 BAC. COUNT=V= -63.421 WEIGHT=W= 60.000
K AND LIND(K) 1 0 2 1 3 1 4 0
OUTPUT FROM SPLIT IS AS FOLLOWS
FOOD NO. 4, TIME= 13, Y= 80.868, V= .095, W= 18.000
FOOD NO. 10, TIME= 13, Y= 80.868, V= .000, W= 34.739

FOOD NUMBER 2 SAUCE (ROUX)

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	5.00000	1
2	5.00000	5.99298	.00047	5.00000	2
3	5.00000	5.93149	.00047	5.00000	3
4	5.00000	5.95563	.00047	5.00000	4
5	5.00000	5.97959	.00047	5.00000	5
6	5.00000	6.03102	.00047	5.00000	6
7	5.00000	6.05524	.00047	5.00000	7
8	5.00000	6.04468	.00047	5.00000	8
9	5.00000	6.00616	.00047	5.00000	9
10	5.00000	6.02130	.00047	5.00000	10
11	5.00000	6.03948	.00047	5.00000	11
12	5.00000	5.90511	.00047	5.00000	12
13	5.00000	5.88074	.00047	5.00000	13

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
14	42.37959	5.90162	.14403	100.00000	14
15	65.05145	2.53755	-6.81470	100.00000	15
16	78.80264	-13.68201	-25.73138	100.00000	16
17	87.14315	-49.75846	-46.26512	100.00000	17

RI= 6.0000 RO= .0000 VE= -49.7585

FOOD NUMBER 3 STOCK AND SEASONING

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	5.00000	1
2	5.00000	5.99296	.00047	5.00000	2
3	5.00000	5.98290	.00047	5.00000	3
4	5.00000	5.97366	.00047	5.00000	4
5	5.00000	5.94438	.00047	5.00000	5
6	5.00000	5.88292	.00047	5.00000	6
7	5.00000	5.93371	.00047	5.00000	7
8	5.00000	5.88202	.00047	5.00000	8
9	5.00000	5.89114	.00047	5.00000	9
10	5.00000	5.91858	.00047	5.00000	10
11	5.00000	5.93891	.00047	5.00000	11
12	5.00000	5.92722	.00047	5.00000	12
13	5.00000	5.94820	.00047	5.00000	13
14	5.00000	6.00450	.00047	5.00000	14
15	5.00000	6.03992	.00047	5.00000	15
16	5.00000	5.97079	.00047	5.00000	16
17	5.00000	5.98977	.00047	5.00000	17

RI= 6.0000 RO= 6.0000 VE= 5.9898
MONREC SIGNAL

FOOD NUMBER 4 EXTRA STOCK

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	80.86755	.09506	-.58988	21.95057	13
2	57.68553	-1.25051	-2.30052	21.95057	14
3	43.62492	-2.34451	.12890	21.95057	15
4	35.09673	-2.24988	.13397	21.95057	16
5	29.92412	-2.16457	.09308	21.95057	17

RI= .0000 RO= .4780 VE= -2.1646

FOOD NUMBER 5 CHILLED VEGETABLES

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	7.00000	.00047	5.00000	1
2	5.00000	7.05232	.00047	5.00000	2
3	5.00000	7.06688	.00047	5.00000	3
4	5.00000	7.09419	.00047	5.00000	4
5	5.00000	7.10400	.00047	5.00000	5
6	5.00000	7.08599	.00047	5.00000	6
7	5.00000	7.03965	.00047	5.00000	7
8	5.00000	7.14487	.00047	5.00000	8
9	5.00000	7.25272	.00047	5.00000	9
10	5.00000	7.32622	.00047	5.00000	10
11	5.00000	7.37216	.00047	5.00000	11
12	5.00000	7.40108	.00047	5.00000	12
13	5.00000	7.34428	.00047	5.00000	13
14	5.00000	7.33122	.00047	5.00000	14
15	5.00000	7.21676	.00047	5.00000	15
16	5.00000	7.17244	.00047	5.00000	16
17	5.00000	7.09677	.00047	5.00000	17

RI= 6.0000 RO= 6.0000 VE= 7.0968

```
MONREC SIGNAL

THIS FOOD ENDS IN BLEND NO.  1. INPUT TO BLEND HAS
  FOOD NO.  2, TIME= 17, Y=  87.143, V= -49.758, WT.=   2.000
  FOOD NO.  3, TIME= 17, Y=   5.000, V=   5.990, WT.=  23.000
  FOOD NO.  4, TIME= 17, Y=  29.924, V=  -2.165, WT.=  18.000
  FOOD NO.  5, TIME= 17, Y=   5.000, V=   7.097, WT.=   8.000
K AND LIND(K) 1  0     2  1     3  1     4  1
OUTPUT FROM BLEND HAS
  FOOD NO.= 6, TIME= 17, Y=  17.018, V=   6.075, WT.=  50.659
BAD SMELL DETECTED
```

FOOD NUMBER 6 MIXTURE ALPHA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	17.01805	6.07520	.01253	98.74610	17
2	49.17553	6.04787	-.19221	98.74610	18
3	68.68003	.84899	-10.30063	98.74610	19
4	80.51011	-18.96976	-29.30058	98.74610	20
5	87.68541	-57.55078	-47.88528	98.74610	21
6	92.03745	-112.62434	-62.27403	98.74610	22
7	94.67710	-179.84624	-72.26360	98.74610	23
8	96.27813	-255.39860	-78.81282	98.74610	24
9	97.24920	-336.20288	-82.97123	98.74610	25

RI= 3.1195 RO= .0000 VE= -336.2029

FOOD NUMBER 7 FROZEN PEAS

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	-18.00000	5.00000	.00000	-18.00000	1
2	-18.00000	5.04637	.00000	-18.00000	2
3	-18.00000	5.07207	.00000	-18.00000	3
4	-18.00000	5.07231	.00000	-18.00000	4
5	-18.00000	5.10884	.00000	-18.00000	5
6	-18.00000	5.19049	.00000	-18.00000	6
7	-18.00000	5.24387	.00000	-18.00000	7
8	-18.00000	5.20385	.00000	-18.00000	8
9	-18.00000	5.21094	.00000	-18.00000	9
10	-18.00000	5.19289	.00000	-18.00000	10
11	-18.00000	5.13368	.00000	-18.00000	11
12	-18.00000	5.06178	.00000	-18.00000	12
13	-18.00000	5.00474	.00000	-18.00000	13
14	-18.00000	4.93552	.00000	-18.00000	14
15	-18.00000	4.96833	.00000	-18.00000	15
16	-18.00000	5.02058	.00000	-18.00000	16
17	-18.00000	5.00300	.00000	-18.00000	17
18	-18.00000	4.91601	.00000	-18.00000	18
19	-18.00000	4.87908	.00000	-18.00000	19
20	-18.00000	4.94492	.00000	-18.00000	20
21	-18.00000	5.03349	.00000	-18.00000	21
22	-18.00000	5.02589	.00000	-18.00000	22
23	-18.00000	5.08801	.00000	-18.00000	23
24	-18.00000	5.07235	.00000	-18.00000	24
25	-18.00000	5.07710	.00000	-18.00000	25

```
RI=    6.0000 RO=    6.0000 VE=    5.0771
MONREC SIGNAL

THIS FOOD ENDS IN BLEND NO.  2. INPUT TO BLEND HAS
  FOOD NO.  6, TIME= 25, Y=  97.249, V=-336.203, WT.=  50.659
```

FOOD NO. 7, TIME= 25, Y= -18.000, V= 5.077, WT.= 6.000
K AND LIND(K) 1 1 2 0 3 0 4 1
OUTPUT FROM BLEND HAS
FOOD NO.= 8, TIME= 25, Y= 85.045, V= 3.003, WT.= 56.370

FOOD NUMBER 8 MIXTURE BETA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	85.04467	3.00321	-.74886	102.24743	25
2	90.71607	-26.17553	-57.63839	102.24743	26
3	94.51773	-90.82529	-71.63224	102.24743	27
4	97.06606	-167.74942	-82.17606	102.24743	28
5	98.77425	-253.75765	-89.79252	102.24743	29
6	99.91929	-346.26971	-95.15236	102.24743	30
7	100.68683	-443.23591	-98.86183	102.24743	31

RI= 3.0000 RO= .0000 VE= -443.2359

FOOD NUMBER 9 SLURRY

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	10.00000	6.00000	.00218	21.64810	25
2	13.84014	6.04010	.00601	21.64810	26
3	16.41427	6.00773	.01098	21.64810	27
4	43.97081	6.00166	.12226	100.00000	28
5	62.44251	3.57010	-4.85747	100.00000	29
6	74.82446	-8.11143	-18.55928	100.00000	30
7	83.12433	-35.05844	-35.37801	100.00000	31

RI= 6.0000 RO= .0000 VE= -35.0584

FOOD NUMBER 10 DE-BONED, COOKED CHICKEN

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	80.86755	.00047	-.58988	5.00000	13
2	69.00678	-5.63722	-10.66191	5.00000	14
3	59.00027	-12.34975	-2.88867	5.00000	15
4	50.55813	-13.97480	-.37139	5.00000	16
5	43.43579	-14.10730	.13203	5.00000	17
6	37.42692	-13.91529	.14540	5.00000	18
7	32.35745	-13.71923	.11399	5.00000	19
8	28.08052	-13.62988	.07693	5.00000	20
9	24.47222	-13.51782	.04830	5.00000	21
10	21.42803	-13.50703	.02965	5.00000	22
11	18.85975	-13.38341	.01836	5.00000	23
12	16.69298	-13.39185	.01168	5.00000	24
13	14.86496	-13.42661	.00770	5.00000	25
14	13.32272	-13.43201	.00529	5.00000	26
15	12.02159	-13.46702	.00379	5.00000	27
16	13.26889	-13.49615	.00522	20.00000	28
17	14.32120	-13.50061	.00676	20.00000	29
18	15.20900	-13.51165	.00835	20.00000	30
19	15.95800	-13.52981	.00991	20.00000	31

RI= .0000 RO= 1.8000 VE= -13.5298

THIS FOOD ENDS IN BLEND NO. 3. INPUT TO BLEND HAS
FOOD NO. 8, TIME= 31, Y= 100.687, V=-443.236, WT.= 56.370
FOOD NO. 9, TIME= 31, Y= 83.124, V= -35.058, WT.= 2.000
FOOD NO. 10, TIME= 31, Y= 15.958, V= -13.530, WT.= 34.739
K AND LIND(K) 1 0 2 1 3 1 4 1
OUTPUT FROM BLEND HAS
FOOD NO.=11, TIME= 31, Y= 68.698, V= .096, WT.= 88.621

FOOD NUMBER 11 MIXTURE GAMMA

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	68.69754	.09593	-.21483	100.60149	31
2	76.96646	-11.21416	-22.22831	100.60149	32
3	83.09223	-39.99038	-35.29872	100.60149	33
4	87.63031	-81.53288	-47.71895	100.60149	34
5	90.99220	-134.64308	-58.58744	100.60149	35
6	93.48275	-197.79175	-67.62117	100.60149	36
7	95.32780	-269.01596	-74.87982	100.60149	37
8	96.69464	-346.69839	-80.57905	100.60149	38
9	97.70723	-429.47360	-84.98229	100.60149	39

RI= .6000 RO= .0000 VE= -429.4736

FOOD NUMBER 12 TOPPING

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	5.00000	6.00000	.00047	97.85880	35
2	19.51710	6.10702	.02088	97.85880	36
3	31.76466	6.20258	.10906	97.85880	37
4	42.09750	6.28747	.14591	97.85880	38
5	50.81495	6.21084	-.41037	97.85880	39

R1= 6.0000 RO= .0000 VE= 6.2108

THIS FOOD ENDS IN BLEND NO. 4. INPUT TO BLEND HAS
FOOD NO. 11, TIME= 39, Y= 97.707, V=-429.474, WT.= 88.621
FOOD NO. 12, TIME= 39, Y= 50.815, V= 6.211, WT.= 5.000
K AND LIND(K) 1 0 2 0 3 1 4 1
OUTPUT FROM BLEND HAS
FOOD NO.=13, TIME= 39, Y= 95.203, V= 4.058, WT.= 87.776

FOOD NUMBER 13 CHICKEN POT PIE

I	Y=TEMP	V=BAC. COUNT	A	YA=AMBIENT TEMP	IAB
1	95.20285	4.05788	-1.19998	23.82914	39
2	82.26499	-13.23796	-33.29614	23.82914	40
3	71.67237	-36.84569	-13.92490	23.82914	41
4	62.99986	-46.39201	-5.23988	23.82914	42
5	55.89942	-49.76034	-1.63052	23.82914	43
6	50.08606	-50.78745	-.30454	23.82914	44
7	45.32649	-50.86390	.08305	23.82914	45
8	33.84734	-50.68722	.12555	-18.00000	46
9	24.44902	-50.54232	.04814	-18.00000	47
10	16.75431	-50.48440	.01183	-18.00000	48
11	10.45443	-50.46622	.00248	-18.00000	49
12	5.29651	-50.38521	.00052	-18.00000	50
13	1.07357	-50.33064	.00012	-18.00000	51
14	-2.38388	-50.32083	.00000	-18.00000	52
15	-5.21460	-50.35106	.00000	-18.00000	53
16	-7.53220	-50.39499	.00000	-18.00000	54
17	-9.42969	-50.34838	.00000	-18.00000	55
18	-10.98323	-50.36465	.00000	-18.00000	56
19	-12.25515	-50.36889	.00000	-18.00000	57
20	-13.29651	-50.43788	.00000	-18.00000	58
21	-14.14911	-50.49345	.00000	-18.00000	59
22	-14.84716	-50.49759	.00000	-18.00000	60
23	-15.41867	-50.48889	.00000	-18.00000	61
24	-15.88659	-50.51970	.00000	-18.00000	62
25	-16.26968	-50.59025	.00000	-18.00000	63
26	-16.58334	-50.76836	.00000	-18.00000	64

27	-16.84014	-50.76234	.00000	-18.00000	65
28	-17.05038	-50.72418	.00000	-18.00000	66
29	-17.22252	-50.72116	.00000	-18.00000	67
30	-17.36345	-50.74397	.00000	-18.00000	68
31	-17.47884	-50.73600	.00000	-18.00000	69
32	-17.57331	-50.71086	.00000	-18.00000	70
33	-17.65066	-50.73888	.00000	-18.00000	71
34	-17.71398	-50.78386	.00000	-18.00000	72
35	-17.76583	-50.82453	.00000	-18.00000	73
36	-17.80828	-50.76659	.00000	-18.00000	74
37	-17.84303	-50.82642	.00000	-18.00000	75
38	-17.87148	-50.77776	.00000	-18.00000	76
39	-17.89478	-50.83903	.00000	-18.00000	77
40	-17.91385	-50.75491	.00000	-18.00000	78
41	-17.92947	-50.71252	.00000	-18.00000	79
42	-17.94225	-50.69732	.00000	-18.00000	80
43	-17.95272	-50.75072	.00000	-18.00000	81
44	-17.96129	-50.70645	.00000	-18.00000	82
45	-17.96831	-50.74403	.00000	-18.00000	83
46	-17.97405	-50.75649	.00000	-18.00000	84
47	-17.97876	-50.76528	.00000	-18.00000	85
48	-17.98261	-50.76587	.00000	-18.00000	86
49	-17.98576	-50.72017	.00000	-18.00000	87
50	-17.98834	-50.71184	.00000	-18.00000	88
51	-17.99045	-50.66400	.00000	-18.00000	89
52	-17.99219	-50.67188	.00000	-18.00000	90
53	-17.99360	-50.71942	.00000	-18.00000	91
54	-17.99476	-50.76701	.00000	-18.00000	92
55	-17.99571	-50.76317	.00000	-18.00000	93
56	16.98848	-50.74446	.01245	175.00000	94
57	45.63111	-50.74464	.07090	175.00000	95
58	69.08171	-56.13640	-10.74591	175.00000	96
59	88.28144	-86.31114	-49.70909	175.00000	97

THIS FOOD IS SERVED AT TIME 97
RI= .0000 RO= .0000 VE= -86.3111
BAD SMELL DETECTED IN STOR. 2 FOOD 5 TIME 13
DURING SPLIT NO. 1 AT END OF FOOD NO. 1
MAX. COUNT IS 7.4011 IN FOOD NO. 5 AT TIME 12
NSIG=1 NSIGD=1 NTRU=1

MONITOR-EFFECTIVENESS SUMMARY
NS= 5 NT= 2 NDO= 0 N11= 2
NSD= 3 NT= 2 ND00= 2 ND11= 2

We make the following comments about this output.

(i) The first main section of the output, ending with the "Intended Schedule of Splits and Blends", is fairly self-explanatory in terms of the variable names described in Appendix II. In the "Intended Schedule of Splits and Blends" the quantities captioned "Proportion", are PR(MS,K), K = 1, 2, ... NJO(MS) and MS is the split number. Also the quantities captioned "Handling Contamination" and "Spillage Weight Loss" are HEVS (MS,K) and HEWS (MS,K), respectively, for K = 1, 2, ... NJO(MS). Similarly for the blends the quantities so-captioned are HEVB(MB) and HEWB(MB).

(ii) In the list of individual food properties following the "Final Schedule", IE is the time at which the food leaves the system. LS = 0 if the food does not end in a split and, if it does, LS is the split number at which the food ends. LB is analogously defined for blends. The lines beginning "K=" give the time, IACH, at which an ambient temperature change occurs, and the new ambient temperature, YAC. One such line is printed for each ambient temperature change, and no lines are printed if the food undergoes no change in ambient temperature. For example, food number 11 suffers no ambient temperature change, while food number 13 experiences two changes.

(iii) In the printout of the results for the various foods, the column headed I is the number of the time-step, starting from the time the food entered the system, while IAB gives the time from the start of the simulation.

(iv) Remember that V, referred to as bacteria count in the program, is really the natural logarithm (i.e. to base e) of the bacterial density. Negative values of V merely mean that the average bacterial density is less than one per gram. When V is large and negative, it means that the probability that any bacteria survives is very low.

(v) Immediately following the table of temperature and bacteria count for each food is information about the monitoring systems and the outcomes of splits occurring in the food. The outcome of a blend is printed following the table for the highest-numbered food entering the blend. If the message "MONREC SIGNAL" is printed, it means that the record-keeping monitor system detected that the food was unsafe (based on the value of TESTI). If the message "BAD-SMELL DETECTED" is printed, it means that the domestic-monitor system detected that the food was unsafe (based on the values of AV1 and SD1). For example, in the first trial of the simulation, a MONREC SIGNAL was obtained for food number 3 because RO = 6 and TESTI = 5, giving RO > TESTI. Also in the second trial the domestic monitor gave a "BAD-SMELL DETECTED" message at blend number 1, following food number 5.

(vi) The quantities K and LIND(K) give information used in the indirectly operating phase of the domestic monitor system. The quantities K are the numbers of storage areas and LIND(K) is an indicator that is 1 if the K-th storage area contains any of the foods involved in the split or blend and 0 if it does not. For example in blend number 1, following food number 5, storage areas 2, 3 and 4 are involved but storage 1 is not. We may verify this by observing that foods 2 to 6 are involved in blend 1, and the Final Schedule shows that foods 2, 3 and 5 are in storage 2, food number 4 is in storage 3, food number 6 in storage 4 and none of the foods is in storage 1.

(vii) Following the end of each trial the maximum bacteria count obtained during that trial is described, and the monitor-system results are summarized. In the first trial, the record-keeping monitor detected a dangerous situation (NSIG=1), the domestic monitor system did not (NSIGD=0) and the situation was not really dangerous (NTRU=0). In other words the record-keeping system gave a false alarm.

(viii) Also, the results of the indirectly-operating phase of the domestic monitor system are described after each trial. Nothing is printed if this monitor system detects nothing, but, if it detects dangerous situations (i.e. a "bad smell"), then messages are printed out describing them. Such a message can be observed at the end of the second trial output above, where we read that during split 1, at time I=13, a bad smell was detected in food number 5, located in storage 2. Referring back to food number 5, we see that V = 7.34 at I = 13, and since AV2 = 7.20, SD2 = 0, the count was high enough to exceed the threshold level, given in this case by AV2. In the first trial, the corresponding value of V is 7.075, which was less than AV2, and therefore no bad smell was detected.

(ix) The monitor-effectiveness summary, given at the very end of the print-out, shows that the record-keeping monitor gave detections in all of the 5 trials (NS=5) that were run, but only in 2 cases was there truly a dangerous situation (NT=2), hence that monitor system gave false-alarms in 3 of the 5 cases tried. The domestic monitor detected danger in 3 out of 5 cases (NSD=3), correctly classifying 2 cases where there was no danger (NDOO=2) and two cases where there was danger (NDII=2), but giving one false alarm.

Appendix IV: Program Listing

The main part of this Appendix is a listing of the Fortran V source program, called BACTRAC, which performs the simulation. There is quite a lot of explanation of the program in this listing by means of comment lines, i.e. lines of text beginning with a C in column 1. Since it is moderately self-explanatory, we give only the following brief comments as guidance:

(i) The heart of the program is the big loop that begins at line 206 and ends at line 464. Each passage through this loop does one trial of the entire simulation. Everything that precedes this big loop is reading in of the data and initialization of some parameters. Within this loop is a smaller (but still large) loop, extending from line 291 to line 420. Each passage through this loop does the calculations for one food, primarily by calling the subroutine BACGRO at line 302.

(ii) The subroutine SPLEND does the calculations involved in the split or blend that occurs after foods have been in storage. It is called twice in the program, at lines 351 and 390. The number IOP determines whether SPLEND does a split (IOP=1) or a blend (IOP=2) calculation.

(iii) The random numbers used in the program are generated by the system subroutine RANDN, which is called at three different places in the program, lines 210, 485 and 543. RANDN(X, N, A, S) generates N random numbers, normally distributed with mean A and variance S^2, and stores them in the array named X. If the program is run on a system that does not have a random number generator of this type in the system subroutines, the user must write one and add it to the subroutines of the program.

```
C          BACTRAC PROGRAM

C***********************************************************************
C
C      THE INSTRUCTIONS OF THE RECORD-KEEPING MONITOR SYSTEM ARE MARKED
C      BY MONREC IN COLUMNS 73-78. THE INSTRUCTIONS OF THE DOMESTIC
C      MONITOR SYSTEM ARE MARKED BY MONDOM IN COLUMNS 73-78.
C      THE INDEX J ALWAYS REFERS TO THE VARIOUS FOODS.
C
C***********************************************************************
      DIMENSION X(140),IC(20),IE(20),W(20),Y(20,250),V(20,250),ICD(20),
     1       WD(20),YD(20),VD(20),    IS(20), IB(20), JSI(20),JBO(20),ISD
    2(20),IBD(20),JSO(20,20),JBI(20,20),      TITLE(12),LEGEND(12,20)
      DIMENSION NJO(20),NJI(20),LS(20),LB(20),YA(20),P(20,6),Q(6)
      DIMENSION IAC(20,5),IACD(20,5),YAC(20,5),NATC(20),LIND(20)
      DIMENSION IO(20),YO(20),YE(20),VE(20),ISER(20),MST(20),YST(20)
      DIMENSION JJ(5),YY(5),VV(5),WW(5),PR(5,5),PRP(5),CONT(20),YSTD(20)
      DIMENSION HEVS(20,5),HEWS(20,5),HEVB(20),HEWB(20),HVS(5),HWS(5)
      DIMENSION IQ(20),YI(20),VI(20),WI(20),IND(20),IACH(20,5)
      DIMENSION IREF(20),VTRAP(20),VREC(20,250),JREC(20),IV(20)
      DIMENSION RI(20),RO(20),LRS(20),LRB(20),JQ(20),IVM(20),VMAX(20)
  200 FORMAT (3H0J=,I2,2X,12A6)
  201 FORMAT (5X,4(I2,2X),2X,3(E12.5,1X),2X,E12.5)
  202 FORMAT(1H0,5X,11HFOOD NUMBER,I3,1X12A6)
  203 FORMAT (1H0,5X,'FINAL SCHEDULE')
```

```
  204 FORMAT(' FOOD',I3,' STARTS AT I=',I3,' AND ENDS AT SPLIT',I3,' STO
     1R. NO.=',I2)
  205 FORMAT(' FOOD',I3,' STARTS AT I=',I3,' AND ENDS AT BLEND',I3,' STO
     1R. NO.=',I2)
  206 FORMAT (' FOOD',I3,' STARTS AT I=',I3,' AND CONTINUES TILL THE END
     1,I=',I3,' STOR. NO.=',I2)
  207 FORMAT (1H0,5X,'INTENDED SCHEDULE OF SPLITS AND BLENDS')
  208 FORMAT ('0SPLIT NUMBER',I2,' IN FOOD NUMBER',I3,' OCCURS AT TIME',
     1I4,' WITH STD DEV ',I3)
  209 FORMAT (' THE NUMBERS AND PROPORTIONS OF THE RESULTING FOODS ARE')
  210 FORMAT (3X,'FOOD NUMBER',I3,'    PROPORTION',F8.5)
  211 FORMAT(1H0,'BLEND',I2,' PRODUCES FOOD',I3,' AT TIME',I4,' WITH S.D
     1. ',I3,' FROM FOODS')
  212 FORMAT (1X,17(I3,1X))
  213 FORMAT (4X,'I',6X,'Y=TEMP',5X,'V=BAC. COUNT',7X,'A',6X,'YA=AMBIENT
     1 TEMP',1X,'IAB')
  214 FORMAT  ('0THIS FOOD ENDS IN SPLIT',I3,' AT TIME, I=',I4,'. INPUT
     1TO SPLIT HAS')
  215 FORMAT (3X,'TEMP=Y=',F8.3,3X'BAC. COUNT=V=',F8.3,' WEIGHT=W=',F8.3
     1)
  216 FORMAT (' OUTPUT FROM SPLIT IS AS FOLLOWS')
  217 FORMAT (3X'FOOD NO. ',I2,', TIME=',I3,', Y=',F8.3,', V=',F8.3,', W
     1=',F8.3)
  218 FORMAT ('0THIS FOOD ENDS IN BLEND NO.',I3,'. INPUT TO BLEND HAS')
  219 FORMAT (3X,'FOOD NO. ',I2,', TIME=',I3,', Y=',F8.3,', V=',F8.3,',
     1WT.=',F8.3)
  220 FORMAT (' OUTPUT FROM BLEND HAS')
  221 FORMAT (3X'FOOD NO.=' I2,', TIME=',I3,', Y=',F8.3,', V=',F8.3,', W
     1T.=',F8.3)
  222 FORMAT (1H1)
  223 FORMAT (' THIS FOOD IS SERVED AT TIME',I3)
  224 FORMAT(' THIS FOOD IS STORED UNTIL THE FINAL TIME,',I4)
  225 FORMAT (' FOOD',I3,' STARTS AT I=',I3,' AND IS SERVED AT I=',I3,'
     1STOR. NO.=',I2)
  226 FORMAT (1X,10(F10.5,1X))
  227 FORMAT (' NJ=',I2,' NMS=',I2,' NMB=',I2,' X(1)=',F7.0,' IMAX=',I4,
     1' SDV=',F9.6,' NREP=',I4)
  228 FORMAT (' IC=',I3,' W=',F9.4,' YI=',F8.4,' VI=',F7.4,' ISER=',I3,'
     1 STOR. NO.=',I2)
  229 FORMAT (' ICD=',I2,' WD=',F8.4,' YD=',F8.4,' VD=',F8.4,' NATC=',I2
     1,' CONT=',F7.4)
  230 FORMAT (' P(J,K)',6(1X,F9.5))
  231 FORMAT (' K=',I2,' IAC=',I3,' IACD=',I2,' YAC=',F9.4)
  232 FORMAT (' K=',I2,' IACH=',I3,' YAC=',F9.5)
  233 FORMAT (' IC=',I3,' IE=',I3,' LS=',I2,' LB=',I2,' NATC=',I2)
  234 FORMAT (' W=',F9.4,' YI=',F9.4,' VI=',F9.4,' YA=',F9.4)
  235 FORMAT ('0STORAGE UNIT',I3,' IS AT TEMP',F8.3,' WITH STD DEV',F8.4
     1)
  236 FORMAT (' BAD SMELL DETECTED')
  237 FORMAT (' BAD SMELL DETECTED JUST BEFORE SERVING')
  238 FORMAT (' AV1=',F8.3,' SD1=',F8.3,' AV2=',F8.3,' SD2=',F8.3)
  239 FORMAT (' BAD SMELL DETECTED IN STOR.',I2,' FOOD',I2,' TIME',I3)
  240 FORMAT (' NSIG=',I1,' NSIGD=',I1,' NTRU=',I1)
  241 FORMAT (' NS=',I3,' NT=',I3,' N00=',I3,' N11=',I3)
  242 FORMAT (' DURING SPLIT NO.',I3,' AT END OF FOOD NO.',I3)
  243 FORMAT (' DURING BLEND NO.',I3,' AT END OF FOOD NO.',I3)
  244 FORMAT (' MAX. COUNT IS',F9.4,' IN FOOD NO.',I3,' AT TIME',I3)
```

```
  245 FORMAT (' NSD=',I3,' NT=',I3,' ND00=',I3,' ND11=',I3)
  246 FORMAT (' HANDLING CONTAMINATION',F10.5,',  SPILLAGE WEIGHT LOSS '
     1,F10.5)
  247 FORMAT (' MONREC SIGNAL')
  248 FORMAT (' NRNO=',I3,' IPTRD=',I1,' NSTOR=',I2)
  249 FORMAT ('     MONITOR-EFFECTIVENESS SUMMARY')
  300 FORMAT (' Q1=',F8.4,' Q2=',F8.4,' Q3=',F8.4,' TEST1=',F8.4,' TRT='
     1,F8.4)
  301 FORMAT (' RI=',F10.4)                                             MONREC
  302 FORMAT (' RI=',F10.4,' RO=',F10.4,' VE=',F10.4 )                  MONREC
  100 FORMAT (12A6)
  101 FORMAT ()
C***********************************************************************
C
C WE READ IN PROPERTIES OF THE FOOD PACKETS
C
C***********************************************************************
      WRITE (6,222)
      READ (5,100) (TITLE (K),K=1,12)
      WRITE(6,100) (TITLE (K),K=1,12)
      READ (5,101) NJ,NMS,NMB,X(1),IMAX,SDV,NREP,NRNO,IPTRD,NSTOR
      WRITE (6,227) NJ,NMS,NMB,X(1),IMAX,SDV,NREP
      WRITE (6,248) NRNO,IPTRD,NSTOR
      READ (5,101)AV1,SD1,AV2,SD2
      WRITE(6,238)AV1,SD1,AV2,SD2
      READ (5,101) Q1,Q2,Q3      ,TEST1,TRT                             MONREC
      WRITE(6,300) Q1,Q2,Q3      ,TEST1,TRT                             MONREC
      DO 10 J=1,NJ
      READ (5,100)   (LEGEND (K,J),K=1,12)
      WRITE (6,200)  J,(LEGEND(K,J),K=1,12)
      READ (5,101) IC(J),W(J),Y(J,1),V(J,1),ISER(J),MST(J)
      WRITE (6,228) IC(J),W(J),Y(J,1),V(J,1),ISER(J),MST(J)
      READ (5,101) ICD(J),        WD(J),YD(J),VD(J),NATC(J),CONT(J)
      WRITE(6,229)ICD(J),        WD(J),YD(J),VD(J),NATC(J),CONT(J)
      READ (5,101) (P(J,K),K=1,6)
      WRITE(6,230)(P(J,K),K=1,6)
      READ (5,101) RI(J)
      WI(J)=W(J)
      YI(J)=Y(J,1)
      VI(J)=V(J,1)
      IQ(J)=IC(J)
      N=NATC(J)
      IF (N.LT.1) N=1
      READ (5,101,ERR=10) (IAC(J,K),IACD(J,K),YAC(J,K),K=1,N)
      DO 46 K=1,N
   46 WRITE (6,231) K,IAC(J,K),IACD(J,K),YAC(J,K)
      IF (IC(J).EQ.1) WRITE (6,301) RI(J)                               MONREC
   10 CONTINUE
      READ (5,101)  (YST(KS),YSTD(KS),KS=1,NSTOR)
      DO 47 KS=1,NSTOR
   47 WRITE (6,235) KS,YST(KS),YSTD(KS)
      WRITE (6,207)
C***********************************************************************
C
C      WE READ IN THE PROPERTIES OF THE SPLITS
C
C***********************************************************************
```

```
      DO 38 J=1,NJ
   38 IND(J)=0
      DO 11 MS=1,NMS
      READ (5,101) NJO(MS)
      N=NJO(MS)
      READ (5,101) IS(MS),JSI(MS),ISD(MS),(JSO(MS,K),PR(MS,K),K=1,N)
      READ (5,101) (HEVS(MS,K),HEWS(MS,K),K=1,N)
      DO 28 K=1,N
      JOY=JSO(MS,K)
   28 IND(JOY)=1
      WRITE (6,208) MS,JSI(MS),IS(MS),ISD(MS)
      WRITE (6,209)
      WRITE (6,210) (JSO(MS,K),PR(MS,K),K=1,N)
      WRITE(6,246) (HEVS(MS,K),HEWS(MS,K),K=1,N)
   11 CONTINUE
C***********************************************************************
C
C     WE READ IN THE PROPERTIES OF THE BLENDS
C
C***********************************************************************
      DO 12 MB=1,NMB
      READ (5,101) NJI(MB)
      N=NJI(MB)
      READ (5,101) IB(MB),JBO(MB),IBD(MB),(JBI(MB,K),K=1,N)
      READ (5,101) HEVB(MB),HEWB(MB)
      JOY=JBO(MB)
      IND(JOY)=1
      WRITE (6,211) MB,JBO(MB),IB(MB),IBD(MB)
      WRITE (6,212) (JBI(MB,K),K=1,N)
      WRITE(6,246) HEVB(MB),HEWB(MB)
   12 CONTINUE
C***********************************************************************
C
C     WE DETERMINE FOR EACH FOOD THE QUANTITIES LS(J) AND LB(J), WHICH
C     SPECIFY WHETHER THE J-TH FOOD ENDS IN A SPLIT OR A BLEND.
C
C***********************************************************************
      DO 13 J=1,NJ
      LS(J)=0
      LB(J)=0
      DO 14 MS=1,NMS
      NKO=NJO(MS)
      DO 15 K=1,NKO
   15 IF (JSI(MS)  .EQ.J) LS(J)=MS
   14 CONTINUE
      DO 16 MB=1,NMB
      NKI=NJI(MB)
      DO 17 K=1,NKI
   17 IF (JBI(MB,K).EQ.J) LB(J)=MB
   16 CONTINUE
   13 CONTINUE
C***********************************************************************
C
C     HERE WE INTRODUCE RANDOMNESS IN INITIAL VALUES AND IN SCHEDULING.
C     ALSO THE BIG LOOP THAT ENDS AT INSTRUCTION NO. 50 DOES THE REPETI-
C     TIONS OF THE ENTIRE SIMULATION.
C
```

```
C**********************************************************************************
      N00=0                                                          MONREC
      N11=0                                                          MONREC
      NS=0                                                           MONREC
      NSD=0                                                          MONDOM
      ND00=0                                                         MONDOM
      ND11=0                                                         MONDOM
      NT=0
      NREF=NMS+NMB                                                   MONDOM
      DO 50 M=1,NREP
      NSIGD=0                                                        MONDOM
      NSIG=0                                                         MONREC
      NTRU=0
      CALL RANDN (X,NRNO,0.,1.)
      DO 29 J=1,NJ
      IF (NATC(J).LT.1) GO TO 43
      N=NATC(J)
      NNR=NMS+NMB+4*NJ+1
      DO 44 K=1,N
      IACH(J,K)=IAC(J,K)+IACD(J,K)*X(NNR)
      NNR=NNR+1
   44 CONTINUE
   43 IF (IC(J).GT.1) GO TO 42
      GO TO 39
   42 IF (IND(J).EQ.1) GO TO 29
      IC(J)=IQ(J)+ICD(J)*X(J+3*NJ)
   39 IA=IC(J)
      W(J)=WI(J)+WD(J)*X(J)
      Y(J,IA)=YI(J)+YD(J)*X(J+NJ)
      V(J,IA)=VI(J)+VD(J)*X(J+2*NJ)
   29 CONTINUE
      DO 32 MS=1,NMS
      N=NJO(MS)
      DO 33 K=1,N
      J=JSO(MS,K)
      IC(J)=IQ(J)+ISD(MS)*X(MS+4*NJ)
   33 CONTINUE
      IS(MS)=IC(J)
      IREF(MS)=IS(MS)                                                MONDOM
   32 CONTINUE
      DO 37 MB=1,NMB
      J=JBO(MB)
      IC(J)=IQ(J)+IBD(MB)*X(NMS+4*NJ+MB)
      IB(MB)=IC(J)
      IREF(NMS+MB)=IB(MB)                                            MONDOM
   37 CONTINUE
      WRITE (6,203)
      DO 34 J=1,NJ
      IF(LS(J).EQ.0) GO TO 35
      WRITE (6,204) J,IC(J),LS(J),MST(J)
      GO TO 34
   35 IF (LB(J).EQ.0) GO TO 36
      WRITE (6,205) J,IC(J),LB(J),MST(J)
      GO TO 34
   36 IF (ISER(J).GE.IMAX) WRITE (6,206) J,IC(J),IMAX,MST(J)
      IF (ISER(J).LT.IMAX) WRITE (6,225) J,IC(J),ISER(J),MST(J)
   34 CONTINUE
```

```
C*******************************************************************************
C
C     HERE WE CALCULATE THE TIME-STEP AT WHICH EACH FOOD TERMINATES
C
C*******************************************************************************
      DO 18 J=1,NJ
      IF (LS(J).LT.1) GO TO 19
      MS=LS(J)
      IE(J)=IS(MS)
      GO TO 21
   19 IF (LB(J).LT.1) GO TO 20
      MB=LB(J)
      IE(J)=IB(MB)
      GO TO 21
   20 IE(J)=IMAX
      IF (ISER(J).LT.IMAX) IE(J)=ISER(J)
   21 WRITE (6,200) J,(LEGEND(K,J),K=1,12)
      IA=IC(J)
      MP=MST(J)
      YA(J)=YST(MP)+X(NNR)*YSTD(MP)
      NNR=NNR+1
      WRITE (6,233) IC(J),IE(J),LS(J),LB(J),NATC(J)
      WRITE (6,234) W(J),Y(J,IA),V(J,IA),YA(J)
      N=NATC(J)
      IF (N.LT.1) GO TO 18
      DO 45 K=1,N
   45 WRITE (6,232)K,IACH(J,K),YAC(J,K)
   18 CONTINUE
C**** **************************************************************************
C
C     HERE BEGINS THE MAIN LOOP,WHICH CALCULATES TEMPERATURE AND BACTER-
C     IAL GROWTH IN EACH FOOD AND THEN MAKES THE CALCULATIONS ARISING
C     OUT OF THE SPLIT OR BLEND AT WHICH THAT FOOD PACKET TERMINATES.
C
C*******************************************************************************
      NNR=NMS+NMB+4*NJ+2
      L=0                                                               MONDOM
      DO 52 J=1,NJ
      II=IC(J)
      DO 23 K=1,6
   23 Q(K)=P(J,K)
      N=NATC(J)
      DO 24 K=1,N
      YO(K)=YAC(J,K)
   24 IO(K)=IACH(J,K)
      WRITE (6,202) J,(LEGEND (K,J),K=1,12)
      IF (IPTRD.NE.1) WRITE (6,213)
      YAA=YA(J)
   41 CALL BACGRO (Q,II,IE(J),IO,YA(J),Y(J,II),YO,YE(J),V(J,II),SDV,NAT
     1C(J),X(NNR),VE(J),IREF,NREF,VTRAP,VMAX(J),IVM(J),CONT(J))
      YA(J)=YAA
      NNR=NNR+1
      DO 54 K=1,NREF                                                    MONDOM
      IZ=IREF(K)                                                        MONDOM
      VREC(J,IZ)=VTRAP(K)                                               MONDOM
   54 CONTINUE                                                          MONDOM
      IF (ISER(J).GE.IMAX) GO TO 48
```

```
      WRITE (6,223) IE(J)
      CALL SNIFF (VE(J),W(J),AV1,SD1,X(NNR),NOSE)                       MONDOM
      IF (NOSE.EQ.1) WRITE (6,237)                                      MONDOM
      NSIGD=NSIGD+NOSE
      NNR=NNR+1
      GO TO 49
   48 IF((LS(J)+LB(J)).EQ.0) WRITE (6,224) IMAX
   49 IT=IE(J)
      Y(J,IT)=YE(J)
      V(J,IT)=VE(J)
      YCOMP=YA(J)
      IF (N.EQ.0) GO TO 57
      DO 56 K=1,N
   56 YCOMP=AMAX1(YCOMP,YO(K))
   57 RO(J)=0.
      IF (YCOMP.LT.Q1) RO(J)=RI(J)
      IF ((YCOMP.GE.Q1).AND.(YCOMP.LE.Q2)) RO(J)=RI(J)+Q3*(YCOMP-Q1)*(IT
     1-II)                                                              MONREC
      WRITE (6,302) RI(J),RO(J),VE(J)                                   MONREC
      IF (RO(J).GE.TEST1) NSIG=NSIG+1                                   MONREC
      IF (RO(J).GE.TEST1) WRITE (6,247)
C
C     FROM HERE TO 25 DESCRIBES WHAT HAPPENS TO A FOOD THAT ENDS IN A
C     SPLIT.
C
      IF (LS(J).EQ.0) GO TO 25
      MS=LS(J)
      N=NJO(MS)
      CALL SNIFF (VE(J),W(J),AV1,SD1,X(NNR),NOSE)                       MONDOM
      NSIGD=NSIGD+NOSE
      NNR=NNR+1
      IOP=1
      WRITE (6,214) MS,IT
      WRITE (6,215,ERR=40) YE(J),VE(J),W(J)
      IF (NOSE.EQ.1) WRITE (6,236)                                      MONDOM
   40 DO 26 K=1,N
      PRP(K)=PR(MS,K)
      HVS(K)=HEVS(MS,K)
      HWS(K)=HEWS(MS,K)
   26 JJ(K)=JSO(MS,K)
      CALL SPLEND (IOP,JSI(MS),N,YE(J),VE(J),W(J),JJ,YY,VV,WW,PRP,X(NNR)
     1,HVS,HWS,NSTOR,MST,LIND)
      NNR=NNR+1
      WRITE (6,216)
      DO 27 K=1,N
      J0=JJ(K)
      RI(J0)=RO(J)                                                      MONREC
      II=IC(J0)
      Y(J0,II)=YY(K)
      V(J0,II)=VV(K)
      W(J0)=WW(K)
   27 WRITE (6,217) J0,II,Y(J0,II),V(J0,II),W(J0)
      GO TO 22
C
C     FROM HERE TO 22 DESCRIBES WHAT HAPPENS WHEN SEVERAL FOODS  END IN
C     A BLEND SIMULTANEOUSLY.
C
```

```
   25 IF (LB(J).EQ.0) GO TO 52
      MB=LB(J)
      N=NJI(MB)
      JMAX=0
      DO 30 K=1,N
   30 JMAX=MAX0 (JMAX,JBI(MB,K))
      IF (J.NE.JMAX) GO TO 52
      IOP=2
      WRITE (6,218) MB
      JU=JBO(MB)
      B=0.                                                              MONREC
      DO 31 K=1,N
      JJ(K)=JBI(MB,K)
      JR=JJ(K)
      B=B+RO(JR)                                                        MONREC
      IT=IE(JR)
      YY(K)=Y(JR,IT)
      VV(K)=V(JR,IT)
      WW(K)=W(JR)
      WRITE (6,219) JR,IT,YY(K),VV(K),WW(K)
   31 CONTINUE
      RI(JU)=B/N                                                        MONREC
      CALL SPLEND (IOP,JU,N,YU,VU,WU,JJ,YY,VV,WW,PRP,X(NNR),HEVB(MB),HEW
     1B(MB),NSTOR,MST,LIND)
      WRITE (6,220)
      NNR=NNR+1
      II=IC(JU)
      Y(JU,II)=YU
      V(JU,II)=VU
      W(JU)=WU
      WRITE(6,221) JU,II,Y(JU,II),V(JU,II),W(JU)
      CALL SNIFF (VU,WU,AV1,SD1,X(NNR),NOSE)                            MONDOM
      NNR=NNR+1
      NSIGD=NSIGD+NOSE
      IF (NOSE.EQ.1) WRITE (6,236)                                      MONDOM
   22 CONTINUE
C
C     THE LOOP ENDING AT 51 DETERMINES THE FOODS CHECKED BY THE DOMESTIC
C     MONITOR SYSTEM DURING THE SPLIT OR BLEND BEING DONE.
C
      DO 51 JIM=1,NJ
      K=MST(JIM)
      IF (LIND(K).EQ.0) GO TO 51
      IF ((II.LE.IC(JIM)).OR.(II.GE.IE(JIM))) GO TO 51
      L=L+1
      IV(L)=II
      JREC(L)=JIM
      LRS(L)=LS(J)
      LRB(L)=LB(J)
      JQ(L)=J
   51 CONTINUE
      NL=L                                                              MONDOM
   52 CONTINUE
C
C     THE LOOP ENDING AT 53 IS ALSO PART OF THE DOMESTIC MONITOR SYSTEM.
C     IT CHECKS ALL FOODS INVOLVED IN THE SPLIT OR BLEND BEING DONE, TO
C     SEE IF THE COOKS SMELL ANYTHING BAD.
```

```
      C
            DO 53 L=1,NL
            I=IV(L)
            J=JREC(L)
            VSM=VREC(J,I)
            CALL SNIFF (VSM,W(J),AV2,SD2,X(NNR),NOSE)
            NNR=NNR+1
            IF (NOSE.NE.1) GO TO 53
            WRITE (6,239) MST(J),J,I
            IF (LRS(L).GT.0) WRITE (6,242) LRS(L),JQ(L)
            IF (LRB(L).GT.0) WRITE (6,243) LRB(L),JQ(L)
            NSIGD=NSIGD+1
         53 CONTINUE
      C
      C     FROM HERE TO 50 WE CALCULATE THE MAX BACTERIA COUNT AND
      C     INFORMATION FOR THE MONITOR-EFFECTIVENESS SUMMARY.
      C
            VMD=0.
            DO 55 J=1,NJ
            IF (VMD.GE.VMAX(J)) GO TO 55
            VMD=VMAX(J)
            JM=J
            IMM=IVM(J)
         55 CONTINUE
            IF (VMD.GE.TRT) NTRU=NTRU+1
            WRITE (6,244) VMD,JM,IMM
            X(1)=X(NNR)*10000000
            IF (NSIG.GE.1) NSIG=1
            IF (NTRU.GE.1) NTRU=1
            IF (NSIGD.GE.1) NSIGD=1
            WRITE (6,240) NSIG,NSIGD,NTRU
            WRITE (6,222)
            NS=NS+NSIG
            NSD=NSD+NSIGD
            NT=NT+NTRU
            IF ((NSIG.EQ.0).AND.(NTRU.EQ.0)) N00=N00+1
            IF ((NSIG.EQ.1).AND.(NTRU.EQ.1)) N11=N11+1
            IF ((NSIGD.EQ.0).AND.(NTRU.EQ.0)) ND00=ND00+1
            IF ((NSIGD.EQ.1).AND.(NTRU.EQ.1)) ND11=ND11+1
         50 CONTINUE
            WRITE (6,249)
            WRITE (6,241) NS,NT,N00,N11
            WRITE (6,245) NSD,NT,ND00,ND11
            STOP
      C*************************************************************************
      C
      C     THIS IS THE END OF THE MAIN PROGRAM, WE NOW START THE LISTING OF
      C     THE SUBROUTINES.
      C     THE SUBROUTINE BACGRO DOES THE CALCULATION OF TEMPERATURE CHANGE
      C     AND BACTERIAL GROWTH FOR EACH FOOD.
      C
      C*************************************************************************
            SUBROUTINE BACGRO (P,I1,IT,IAC,YG,YC,YAD,YU,VC,SDV,NC,SEED,VU,IREF
           1,NREF,VTRAP,VM,IM,Z)
            DIMENSION P(6),IAC(5),YAD(5),Y(100),V(100),A(100),IR(5),X(100)
           1,ITRAP(20),IREF(20),VTRAP(20)
        101 FORMAT (2X,I3,2X,4(F12.5,2X),I3)
```

```
      X(1)=SEED*100000
      NI=IT-I1+1
      IR(NC+1)=NI
      CALL RANDN (X,NI,Z,SDV)
      IF (NC.EQ.0) GO TO 5
      DO 4 K=1,NC
    4 IR(K)=IAC(K)-I1+1
    5 CONTINUE
      DO 6 K=1,NREF                                                     MONDOM
      VTRAP (K)=0.                                                      MONDOM
    6 ITRAP(K)=IREF(K)-I1+1                                             MONDOM
      I=1
      Y(1)=YC
      IF(Y(1).LT.P(2)) A(1)=0.
      IF (Y(1).GT.P(5)) A(1)=P(3)-P(6)*(Y(1)-P(5))**2
      IF ((Y(1).GE.P(2)).AND.(Y(1).LE.P(5))) A(1)=P(3)*EXP(-P(4)*(P(5)-Y
     1(1))**2)
      V(1)=VC
      I0=1
      IF (IPTRD.NE.1) WRITE (6,101) I,Y(1),V(1),A(1),YG,I1
      IM=0
      VM=0.
      DO 1 I=2,NI,1
      IF(NC.EQ.0) GO TO 3
      DO 2 K=1,NC
      IF ((I.LE.IR(K)).OR.(I.GT.IR(K+1))) GO TO 2
      I0=IR(K)
      YC=Y(I0)
      YG=YAD(K)
    2 CONTINUE
    3 Y(I)=YG+(YC-YG)*EXP((I0-I)*P(1))
      IF(Y(I).LT.P(2)) A(I)=0.
      IF (Y(I).GT.P(5)) A(I)=P(3)-P(6)*(Y(I)-P(5))**3
      IF ((Y(I).GE.P(2)).AND.(Y(I).LE.P(5))) A(I)=P(3)*EXP(-P(4)*(P(5)-Y
     1(I))**2)
      V(I)=V(I-1)+.5*(A(I-1)+A(I))
      V(I)=V(I)+X(I)
      IF (VM.LE.V(I)) IM=I+I1-1
      VM=AMAX1(VM,V(I))
      IAB=I1+I-1
      IF (IPTRD.NE.1) WRITE (6,101) I,Y(I),V(I),A(I),YG,IAB
      DO 7 K=1,NREF                                                     MONDOM
    7 IF (I.EQ.ITRAP(K)) VTRAP(K)=V(I)                                  MONDOM
    1 CONTINUE
      YU=Y(NI)
      VU=V(NI)
      RETURN
C***********************************************************************
C
C     THE SUBROUTINE SPLEND CARRIES OUT THE CALCULATIONS ARISING FROM
C     EITHER THE SPLIT OR THE BLEND AT WHICH A FOOD PACKET ENDS.
C
C***********************************************************************
      SUBROUTINE SPLEND (IOP,JU,NU,YU,VU,WU,J,Y,V,W,FR,SEED,HV,HW,NSTOR,
     1MST,LIND)
      DIMENSION J(20),Y(20),W(20),V(20),FR(20),X(50),HV(20),HW(20),MST(2
     20),LIND(20)
```

```
      101 FORMAT ()
      200 FORMAT (' K AND LIND(K)',10(I2,2X,I2,4X))
          X(1)=SEED*100000
          NO=2*NU+2
          CALL RANDN (X,NO,0.,1.)
          GO TO (10,20),IOP
       10 DO 1 K=1,NU
          Y(K)=YU
          IF (VU.LT.0.) VU=0.
          V(K)=VU+HV(K)*ABS(X(K))
          W(K)=WU*FR(K)*(1.-HW(K)*ABS(X(K+NU)))
        1 CONTINUE
          GO TO 30
       20 WU=0.
          XU=0.
          YU=0.
          DO 2 K=1,NU
          WU=WU+W(K)
          XU=XU+W(K)*EXP(V(K))
          YU=YU+W(K)*Y(K)
        2 CONTINUE
          YU=YU/WU
          VU=ALOG(XU/WU)
          IF (VU.LT.0.) VU=0.
          VU=VU+HV(1)*ABS(X(1))
          WU=WU*(1.-HW(1)*ABS(X(2)))
       30 CONTINUE
C
C         FROM HERE TO THE END OF SPLEND THE QUANTITY LIND(K) IS FOUND. THIS
C         IS PART OF THE DOMESTIC MONITOR SYSTEM. LIND(K)=1 IF THE K-TH
C         STORAGE IS INVOLVED IN THE SPLIT OR BLEND BEING DONE, OTHERWISE
C         LIND(K)=0.
C
          DO 3 K=1,NSTOR
        3 LIND(K)=0
          M1=MST(JU)
          LIND(M1)=1
          DO 4 K=1,NU
          J1=J(K)
          M1=MST(J1)
        4 LIND(M1)=1
          WRITE (6,200) (K,LIND(K),K=1,NSTOR)
          RETURN
C***************************************************************************
C
C         THE SUBROUTINE SNIFF CARRIES OUT THE DETERMINATION OF WHETHER THE
C         COOKS PERCEIVE THAT SOMETHING IS WRONG WITH THE FOOD THEY ARE
C         WORKING ON. THIS IS PART OF THE DOMESTIC MONITOR SYSTEM.
C
C***************************************************************************
          SUBROUTINE SNIFF (VT,WT,AV,SD,XX,KSM)
          KSM=0
          C=AV+SD*XX
          IF (VT.GT.C) KSM=1
          RETURN
          END
```

Appendix V: Executive Control Cards

In describing the control cards needed to run this simulation on the Natick Laboratories UNIVAC 1106 system, we shall begin by assuming the following:

(i) The program BACTRAC is stored in a file named MIMOTAPE on magnetic tape reel number 0513.

(ii) The user wishes to enter a new set of input data, which is now stored on cards and which will become the element ELNAME after it is stored in the memory.

Then the arrangment of control cards is as follows, where we are using ϕ to stand for the number zero:

Control Card 1: @ RUN NLMIMO, GSDϕ3ϕT, FTLEESIM

Control Card 2: @ ASG,T MIMOTAPE., 6C, ϕ513W

Control Card 3: @ ASG,PU MIM.

Control Card 4: @ COPIN MIMOTAPE., MIM.

Control Card 5: @ ELT,ILS MIM.ELNAME

The data deck follows control card 5.

Control Card 6: @ FOR, US MIM.BACTRAC, TPF$.MOD

Control Card 7: @ XQT

Control Card 8: @ ADD, PE MIM.ELNAME

Control Card 9: @ PACK MIM.

Control Card 10: @ PRT,T MIM.

Control Card 11: @ FIN

On these and subsequent control cards, ELNAME and MIM can be replaced wherever they occur by any other descriptive names, having twelve or fewer letters, that the user prefers.

The foregoing control cards will cause the input deck to be listed on the printer, then used as the data for the BACTRAC program. This data deck (i.e. ELNAME) is added to the contents of the file MIM. and retained in the computer memory.

The most common situation is that the user will need to make several computer runs, either because he wants to see the simulation results under different conditions or because the input data deck has errors in it that have to be corrected. The arrangement of the control cards on subsequent runs is similar to that of the first run (described above) with the following changes:

(a) Cards 2 and 4 are removed

(b) Card 3 is replaced by
@ ASG,A MIM.

(c) On card 5 replace I by U in column 6

(d) Immediately following card 5, instead of the data cards are the correction cards, which the user must prepare following the instructions in the UNIVAC 1100 Operating System Manual, Section 9.5. These cards cause the computer to make changes in the information stored in ELNAME and print its revised version. The program is then executed using the revised data.

When the user is finally finished using the program, he must remove it from the computer memory. This is done by means of the following control cards

Control Card 1: as before

Control Card 2: @ ASG,A MIM.

Control Card 3: @ DELETE MIM.

Control Card 4: @ PRT,T MIM.

Control Card 5: @ FIN

If a user meets difficulties in running this simulation on the Natick Laboratories UNIVAC 1106 system, he should consult the author of this report or members of the Data Analysis Office. In attempting to run the simulation on a different computer system the user should first consult with people who are familiar with the system he plans to use. The executive control cards described above will be irrelevant in that case.