Answer Booklet

BASIC Computer Programming
A Self-Instructional Course

Approved for public release; distribution unlimited
Instructions

The answers for the problems given in the test booklet are coded in the following manner:

There are four phases in the course — I, II, III, and IV. You will notice that the pages of the text booklet are marked at the top of each page with the appropriate phase number. Within the individual phases, there are parts, such as Part One, Part Two, etc. As the problems are given in the text material, they are numbered according to the part in which they appear. For example, on a page marked III at the top, a problem numbered 3.2 would be Problem 2 of Part Three of Phase III.

In this Answer Booklet, the answers to the problems are coded exactly the same way. The phase number will be at the top of each page. The answers will be listed, for example, as: Problem 3.2. (meaning that this is the answer to Problem 2 of Part Three of Phase _ _).

Approved for public release; distribution unlimited
ANSWERS TO PROBLEMS OF PHASE I: BASIC OPERATIONS

Problem 3.1.
CLA EENY
ADD MEENY
ADD MING

Problem 3.2.
CLA LEAVE
ADD AMOL
ADD TDY
STR ABSENT

Problem 3.3.
CLA VALUE
ADD VALUE
STR VALUE

Problem 3.4.
CLA REGPAY
ADD OVTPAY
SUB SOCSEC
SUB INCTAX
STR NETPAY

Problem 3.5.
CLA PENCIL
SUB ERASER
SUB ERASER
STR PEN

Problem 3.6.
CLA RCVD
ADD STOCK
SUB ISSUE1
SUB ISSUE2
SUB ISSUE3
STR STOCK
HLT

Problem 3.7.
CLA REGPAY
ADD OVTPAY
STR TOTPAY
SUB DEDUCTION
STR NETPAY
HLT

Problem 3.8.
CLA EW
ADD ENCOME
SUB ENGO
STR EM
CLA EM
ADD ENCOME
SUB ENGO
STR EM
HLT

Problem 3.9.
CLA RECORD
STR X
ADD RECORD
STR Y
ADD RECORD
STR Z
HLT
<table>
<thead>
<tr>
<th>Problem 4.1</th>
<th>Problem 4.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA RECORD</td>
<td>CLA EMPLOY+1</td>
</tr>
<tr>
<td>ADD RECORD+1</td>
<td>ADD EMPLOY+4</td>
</tr>
<tr>
<td>SUB RECORD+2</td>
<td>ADD EMPLOY+7</td>
</tr>
<tr>
<td>STR RECORD+3</td>
<td>STR LOYAL</td>
</tr>
<tr>
<td>HLT</td>
<td>HLT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA WORKER</td>
</tr>
<tr>
<td>STR WORKER+1</td>
</tr>
<tr>
<td>CLA SALARY</td>
</tr>
<tr>
<td>STR SALARY+1</td>
</tr>
<tr>
<td>CLA TAXES</td>
</tr>
<tr>
<td>STR TAXES+1</td>
</tr>
<tr>
<td>HLT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 4.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA MAN</td>
</tr>
<tr>
<td>STR WORKER</td>
</tr>
<tr>
<td>CLA MAN+1</td>
</tr>
<tr>
<td>STR WORKER+1</td>
</tr>
<tr>
<td>CLA MAN+2</td>
</tr>
<tr>
<td>STR WORKER+2</td>
</tr>
<tr>
<td>HLT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 4.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA STOCK</td>
</tr>
<tr>
<td>ADD GET</td>
</tr>
<tr>
<td>SUB ISSUE</td>
</tr>
<tr>
<td>STR STOCK+1</td>
</tr>
<tr>
<td>ADD GET+1</td>
</tr>
<tr>
<td>SUB ISSUE+1</td>
</tr>
<tr>
<td>STR STOCK+2</td>
</tr>
<tr>
<td>ADD GET+2</td>
</tr>
<tr>
<td>SUB ISSUE+2</td>
</tr>
<tr>
<td>STR STOCK+3</td>
</tr>
<tr>
<td>HLT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 4.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA WORKER</td>
</tr>
<tr>
<td>STR WORKER+1</td>
</tr>
<tr>
<td>CLA SALARY</td>
</tr>
<tr>
<td>STR SALARY+1</td>
</tr>
<tr>
<td>CLA TAXES</td>
</tr>
<tr>
<td>STR TAXES+1</td>
</tr>
<tr>
<td>HLT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA PAYRT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>STR GROPAY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA BAGNNO</td>
</tr>
<tr>
<td>STR OUTPUT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 5.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA RETIRE</td>
</tr>
<tr>
<td>STR OVTRET</td>
</tr>
</tbody>
</table>
Problem 5.5.
CLA SPACE
STR LOC1
STR LOC2
STR LOC3

Problem 5.6.
CLA SPACE
ADD VERB
ADD VERB
ADD QUANT
ADD QUANT
ADD QUANT
STR MOS1
HLT

Problem 5.7.
CLA RECPAY
ADD OVPAY
SUB DEDUCT
STR NETPAY
HLT

Problem 5.8.
CLA PAY
ADD PAY
STR PAY
STR RECORD+18
HLT

Problem 5.9.
CLA EMPLOY
STR OUTPUT
CLA EMPLOY+5
STR OUTPUT+1
CLA EMPLOY+6
STR OUTPUT+2
HLT

Problem 5.10.
CLA EMPLOY+1
ADD EMPLOY+2
ADD EMPLOY+3
ADD EMPLOY+4
STR DEDUCT
HLT

Problem 5.11.
CLA EMPLOY+5
SUB EMPLOY+1
SUB EMPLOY+2
SUB EMPLOY+3
SUB EMPLOY+4
STR EMPLOY+6
HLT

-or-
CLA EMPLOY+5
SUB DEDUCT
STR EMPLOY+6
HLT

Problem 5.12.
CLA REGTIM
ADD OVRTIM
ADD BONUS
SUB EMPLOY+1
SUB EMPLOY+2
SUB EMPLOY+3
SUB EMPLOY+4
STR EMPLOY+5
HLT
Problem 5.13.

CLA EMPLOY
STR EMMUM
CLA REGTIM
STR TOTPAY
CLA OVRTIM
STR TOTPAY+1
CLA BONUS
STR TOTPAY+2
CLA EMPLOY+1
STR DEDUCT
CLA EMPLOY+2
STR DEDUCT+1
CLA EMPLOY+3
STR DEDUCT+2
CLA EMPLOY+4
STR DEDUCT+3
CLA EMPLOY+5
STR NETPAY
HLT

Problem 5.14.

CLA AMMO
SUB HEUSED
STR AMMO
CLA AMMO+1
SUB APUSED
STR AMMO+1
CLA AMMO+2
SUB CMUSED
STR AMMO+2
CLA AMMO+3
SUB PRUSED
STR AMMO+3
HLT
ANSWERS TO PROBLEMS OF PHASE II: BASIC LOOPING

Problem 1.1.

CLA COST
REPEAT ADD TRANS
TRU REPEAT

The problem asks for the sum of COST+TRANS+TRANS+TRANS+etc. We can get one COST by writing CLA COST; but to add more than one TRANS it is necessary to transfer back to the ADD TRANS instruction again and again. TRU REPEAT does just that, transferring the computer to REPEAT, which names ADD TRANS.

As a result, a loop or circle is formed from TRU REPEAT back to ADD TRANS, giving the endless series: ADD TRANS, TRU REPEAT, ADD TRANS, TRU REPEAT, ADD TRANS, and so on. The loop is diagramed below:

<table>
<thead>
<tr>
<th>CLA COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPEAT</td>
</tr>
<tr>
<td>ADD TRANS</td>
</tr>
<tr>
<td>TRU REPEAT</td>
</tr>
</tbody>
</table>

the loop:

Problem 1.2.

CLA PANTS
REPEAT ADD SUITS
TRU REPEAT

Problem 1.3.

AGAIN CLA COUNT
SUB ONE
STR COUNT
TRU AGAIN

There are two points to notice in this program. The first is the STR COUNT instruction, which erases the contents of COUNT and copies in the contents of the accumulator. Thus, if COUNT starts with a 5, SUB ONE brings it down to 4, and STR COUNT erases the 5 and copies in a 4.

Secondly, the symbolic location AGAIN is placed beside CLA COUNT so the instructions will be repeated starting with CLA COUNT. Notice that if COUNT started with a 5 before the first loop, it will have a 4 at the start of the second loop.
Problem 1.4.

CLA LOOPS
AGAIN SUB ONE OR
STR LOOPS TRU AGAIN

Problem 1.5.

CLA ZRO
REPEAT ADD COST
TRU REPEAT

Notice what would happen if the accumulator were not zeroed with CLA ZRO. The ADD command adds the number from the location addressed (in this case, COST) to whatever is already in the accumulator. If the accumulator started with some number from a previous program and we did not clear it out with CLA ZRO, the program would add that previous number to all the COSTs.

Problem 1.6.

CLA ZRO
AGAIN ADD HAT
TRU AGAIN

Problem 1.7.

CLA ZRO
LOOPER ADD HAT
ADD COAT
ADD GLOVE
TRU LOOPER

Problem 2.1.

CLA COUNT
SUB ONE
STR COUNT
Problem 2.2.

CLA LOOPER
SUB ONE
STR LOOPER

Problem 2.3.

LOWER CLA CASE
SUB ONE
STR CASE
TRU LOWER

Problem 2.4.

AGAIN CLA CARD
ADD ONE
STR CARD

CLA FILE
SUB ONE
STR FILE
TRU AGAIN

adding a 1 to CARD
subtracting a 1 from FILE
the loop instruction

This program can be thought of in terms of three component blocks: one block to add a 1 to CARD, a second block to subtract a 1 from FILE, and a final instruction for looping.

Problem 2.5.

REPEAT CLA COUNT
SUB ONE
SUB ONE
STR COUNT
TRU REPEAT

The clue to this problem is in understanding how the SUB command works. It changes the number in the accumulator, leaving the changed number in the accumulator for the next instruction. Or think of it this way: You have repeated the ADD command several times in a row; the SUB command can be treated the same way.
Problem 3.1.

CLA COUNT
SUB ONE
STR COUNT
TRZ STOP

Incidentally, the program would work in exactly the same way if TRZ STOP were written as the third instruction, with STR COUNT fourth. The reason is that neither TRZ nor STR disturbs the contents of the accumulator. We will keep placing TRZ STOP fourth simply as a matter of convenience.

Problem 3.2.

CLA LOOPS
SUB ONE
STR LOOPS
TRZ STOP

Problem 3.3.

CLA HATS1
TRZ STOP
STR HATS2
STOP HLT

Programmers often work out their programs using "flow charts." Problem 3.3 is shown as a flow chart below:

```
Bring up HATS1
Does HATS1 have a zero?
  Yes
  No
  Store in HATS2
  HLT
```
Problem 3.4.

REPEAT CLA FILE
    ADD ONE
    STR FILE
    CLA CARD
    SUB ONE
    STR CARD
    TRZ STOP
    TRU REPEAT
STOP HLT

Problem 3.5.

DOWN CLA MANY
    SUB ONE
    STR MANY
    TRZ STOP
    TRU DOWN
STOP HLT

Problem 3.6.

LOWER CLA DECK
    SUB UNIT
    STR DECK
    TRZ STOP
    TRU LOWER
STOP HLT

Problem 4.1.

CLA ONCE
    ADD MANY
    STR MANY

The program adds the basic number (in ONCE) to the answer location (MANY), storing the sum back in the answer location.

Problem 4.2.

CLA LACE
    ADD SHOE
    STR SHOE
II

Problem 4.3.

CLA ZRO
STR TOTAL

REPEAT
CLA PAYOFF
ADD TOTAL
STR TOTAL
TRU REPEAT

Cleaning out garbage

Adding

Problem 4.4.

CLA ZRO
STR VALUE

REPEAT
CLA SUIT
ADD VALUE
STR VALUE
TRU REPEAT

Problem 4.5.

CLA ZRO
STR TOTAL

REPEAT
CLA SALARY
ADD TOTAL
STR TOTAL

CLA MEN
SUB ONE
STR MEN
TRZ STOP

TRU REPEAT

Cleaning out garbage

Adding

Test for completion

Looping

STOP HLT
Problem 4.6.

CLA ZRO
STR GROSS
AGAIN CLA PRICE
ADD GROSS
STR GROSS
CLA SALES
SUB ONE
STR SALES
TRZ STOP
TRU AGAIN
STOP HLT

Problem 5.1.

CLA SOLD2
TRZ STOP
STR COUNT

Program preparation

CLA COUNT
SUB ONE
STR COUNT
TRZ STOP

Test for completion

Problem 4.7.

CLA ZRO
STR VALUE
REPEAT CLA PRICE
ADD VALUE
STR VALUE
CLA PENCIL
SUB K1
STR PENCIL
TRZ STOP
TRU REPEAT
STOP HLT

Problem 5.2.

CLA AGE
TRZ STOP
STR OLD

Program preparation

CLA OLD
SUB ONE
STR OLD
TRZ STOP

Test for completion

Problem 4.8.

CLA ZRO
STR ALL
MULT CLA THREE
ADD ALL
CLA MEN
SUB UNIT
STR MEN
TRZ STOP
TRU MULT
STOP HLT

Problem 5.3.

CLA COUNT
TRZ STOP
STR TEMP

Checking and saving the loop counter

CLA ZRO
STR VALUE

Cleaning out garbage

CLA COST
ADD VALUE
STR VALUE

Adding

CLA TEMP
SUB ONE
STR TEMP
TRZ STOP

Test for completion

TRU REPEAT

Looping

STOP HLT

STOP HLT
Problem 5.4.

CLA HAND+1
TRZ STOP
STR HAND+2
CLA ZRO
STR HAND+3
WASH
CLA HAND
ADD HAND+3
STR HAND+3
CLA HAND+2
SUB KON1
STR HAND+2
TRZ STOP
TRU WASH
STOP HLT

Problem 6.1.

Program C

Problem 6.2.

CLA SOLD
TRZ STOP
STR TEMP
CLA ZRO
STR TOTAL
SELL
CLA BIG
ADD LITTLE
ADD TOTAL
STR TOTAL
CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
TRU SELL
STOP HLT

This is the only tricky part.
Problem 6.3.

CLA SOLD
TRZ STOP
STR COUNT
CLA HRRNS
STR BLOW
You should save HRRNS by relocating it into
BLow before zeroing out HRRNS, used later
as the answer location.

CLA ZRD
STR HRRNS
POST CLA PRICE
ADD HRRNS
STR HRRNS
CLA COUNT
SUB ONE
STR COUNT
TRZ STOP
TRU POST
STOP HLT

Problem 6.4.

CLA ASK
STR COUNT
CLA ZRD
STR TOTAL
FIRST CLA TUBE
ADD TOTAL
1st Program
STR TOTAL
CLA COUNT
SUB CON (Not ONE)
STR COUNT
TRZ SECON
TRU FIRST

SECOND CLA ASK(+)
STR COUNT+1
THIRD CLA TUBE+1
ADD TOTAL
STR TOTAL
2nd Program
CLA COUNT+1
SUB CON
STR COUNT+1
TRZ STOP
TRU THIRD
STOP HLT

Since you cleaned out garbage
from the answer location TOTAL
during program preparation for
the first program, you needn't
do it again for the second.
Problem 6.5.

Do not zero out location AVAIL; you need that number.

1st test for completion; it uses TEMP, not MEN.

Subtraction: AVAIL - MEAL2 = AVAIL

2nd test for completion; it uses VISIT, not VIP.

Program preparation for the second program could have been accomplished along with preparation for the first, eliminating symbolic location NEXT.
Problem 6.6.

Use NEXT, not STOP.

Zero out both answer locations at the beginning; it saves an instruction.

1st test for completion, transferring to the second program.

If there's nothing, go to BOTH.

2nd test for completion, transferring to final instruction.

Symbolic location STOP isn't needed at all.
Problem 6.7.

You don't need to save either number so leave them in the same locations, transferring to the second program if either is zero. Remember CLA does not change storage loc.

Zero both answer locations here.

This routine computes $FIRST = COST \times ITEMS$ by adding COST into FIRST as many times as ITEMS.

Transfer to the second program.

These instructions say, "Transfer to final computations if either A or B equals zero."

This routine computes $SECOND = A \times B$, using B as the loop counter.

Since $FIRST = COST \times ITEMS$, and $SECOND = A \times B$, it follows that $(COST \times ITEMS) - (A \times B) = (FIRST - SECOND)$.

That was a tough problem. Congratulations if you got it all right.
ANSWERS TO PROBLEMS OF PHASE III: DATA PROCESSING

Problem 1.1.

CLA REPEAT
ADD ONE
STR REPEAT

Problem 1.2.

CLA HUBERT
ADD DIGIT
STR HUBERT

Problem 1.3.

CLA LOWER
ADD ONE
ADD ONE
STR LOWER

Problem 1.4.

CLA ZRO
STR FINAL
REPEAT CLA VALUE
ADD FINAL
STR FINAL
CLA REPEAT
ADD K
STR REPEAT
TRU REPEAT
Change VALUE to VALUE+1 so that on the next loop VALUE+1 will be added into FINAL.
III

Problem 1.5.

CLA ZRO
STR EVERY
AGAIN CLA COST
ADD EVERY
STR EVERY
CLA AGAIN
ADD TWO
STR AGAIN
TRU AGAIN

Problem 1.6.

CLA MEN
STR TEMP
CLA ZRO
STR TOTAL
SELL CLA SALES
ADD TOTAL
STR TOTAL
CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
CLA SELL
ADD ONE
STR SELL
TRU SELL
STOP HLT

COST is modified to COST+2 for the
next loop.

(There is no need for a TRZ STOP instruction since 16 is obviously greater than 0.)

Add in the number of sales made by the
first salesman.

The test for completion.

Get the address for the second salesman
ready.

Start the loop again to add in the sales
made by the second salesman.
Problem 1.7.

CLA TYPES  STR POP
CLA ZRO  STR SALES
HIT  CLA RECORD  ADD SALES  STR SALES
CLA POP  SUB K1  STR POP  TRZ STOP
CLA HIT  ADD K1  STR HIT
TRU HIT
STOP  HLT

Problem 1.8.

CLA ITEMS  TRZ STOP  STR HOLD  CLA ZRO  STR HOURS
REPEAT  CLA TUBE+3  ADD HOURS  STR HOURS
CLA HOLD  SUB ONE  STR HOLD  TRZ STOP
CLA REPEAT  ADD FOUR  STR REPEAT
TRU REPEAT
STOP  HLT

(You have been told there is at least one type.)

The 4th word has the hours the tube was used, which is TUBE+3.

Since each record is four words long, we add a 4 to get to the corresponding word in the next record.
Problem 1.9.

CLA HELP
TRZ STOP
STR TEMP
CLA ZRO
STR REGPAY

TOTAL
CLA RECORD+2
ADD REGPAY
STR REGPAY
CLA TEMP
SUB CON1
STR TEMP
TRZ STOP
CLA TOTAL
ADD CON4
STR TOTAL
TRU TOTAL
STOP
HLT

Problem 2.1.

CLA ZRO
STR BONUS

CHECK
CLA PUSH
TRZ TEST
CLA BONUS
ADD UNIT
STR BONUS

TEST
CLA MEN
SUB UNIT
STR MEN
TRZ STOP

CLA CHECK
ADD UNIT
STR CHECK

TRU CHECK

STOP
HLT
Problem 2.2.

CLA MEDICS
STR TEMP
CLA ZRO
STR CALLS
AGAIN CLA DOCTOR
TRZ OUT
CLA CALLS
ADD ONE
STR CALLS
OUT CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
CLA AGAIN
ADD ONE
STR AGAIN
TRU AGAIN
STOP HLT

CLA MEDICS
STR TEMP
CLA ZRO
STR CALLS
AGAIN CLA DOCTOR
TRZ OUT
if not zero
CLA CALLS
ADD ONE
STR CALLS
if zero
CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
CLA AGAIN
ADD ONE
STR AGAIN
TRU AGAIN
STOP HLT
Problem 2.3.

Program preparation

REPEAT CLA TAXES TRZ GETHIM
TEST test for completion
    address modification
    TRU REPEAT
GETHIM CLA NOTYET ADD K1
    STR NOTYET TRU TEST
STOP HLT

If the number is zero, the man has not paid his taxes, and should be counted.

Non-taxpayers are counted here. When these instructions are performed, they come immediately after the transfer instruction; since this means skipping the test for completion and address modification, a transfer instruction (TRU TEST) is needed to go back and perform them before starting another loop.

Problem 2.4.

CLA PERSON TRZ STOP
    STR FILE
    CLA ZRO
    STR EM
REPEAT CLA NUMBER+2 TRZ ENLIST
FINAL CLA FILE
    SUB K1
    STR FILE
    TRZ STOP
    CLA REPEAT ADD K4
    STR REPEAT
    TRU REPEAT
ENLIST CLA EM
    ADD K1
    STR EM
    TRU FINAL
STOP HLT

A zero indicates an enlisted personnel; transfer out to count him.

Test for completion.

Address modification.

Count enlisted personnel.

You have transferred out of the loop, so you must get back in.
Problem 2.5.

CLA FIRM
TR2 STOP
STR SAVE
CLA ZRO
STR SINGLE

CYCLE CLA WORKER+3
TR2 FREE
DONE CLA SAVE
SUB ONE
STR SAVE
TR2 STOP
CLA CYCLE
ADD SIX
STR CYCLE
TRU CYCLE
FREE CLA SINGLE
ADD ONE
STR SINGLE
TRU DONE

STOP HLT

Continue the loop where you transferred out.
Problem 2.6.

CLA PLANES
STR FLY
CLA ZRO
STR ONTIME
STR LATE

CHECK CLA FLIGHT+4
TRZ GOOF
CLA ONTIME
ADD ONE
STR ONTIME

TEST CLA FLY
SUB ONE
STR FLY
TRZ STOP

CLA CHECK
ADD FIVE
STR CHECK

TRU CHECK

GOOF CLA LATE
ADD ONE
STR LATE

TRU TEST

STOP HLT

Zero out both answer locations.

Zeroes indicate late flights; transfer out to count them.

If a flight wasn't late (a 0), it was on time; count it here.

All loops must end with a test for completion and address modification, whether they involve a transfer or not.

Start the next loop.

Count late flights here.

Go back and finish the loop.
Problem 2.7.

CLA TYPES
STR COUNT
CLA ZRO
STR YES
STR NO

LOOP: CLA PAINT+2
       TRZ NOSOL
       CLA YES
       ADD CON1
       STR YES

LAST: CLA COUNT
       SUB CON1
       STR COUNT
       TRZ STOP
       CLA LOOP
       ADD CON4
       STR LOOP
       TRU LOOP

NOSOL: CLA NO
       ADD CON1
       STR NO
       TRU LAST

STOP: HLT
Problem 3.1.

CLA TOTAL
TRZ STOP
STR NUMBER
CLA ZRO
STR MEDIUM
REPEAT CLA BRA
SUB TWO
TRZ COUNT
LAST CLA NUMBER
SUB ONE
STR NUMBER
TRZ STOP
CLA REPEAT
ADD ONE
STR REPEAT
TRU REPEAT
COUNT CLA MEDIUM
ADD ONE
STR MEDIUM
TRU LAST
STOP HLT

If the number was: A B C D
it is now: 1 0 1 2 .
A zero now marks a B-bra; transfer out
of the loop to count them.

The test for completion.

Modify BRA to BRA+1.

Start the loop again.

Add a 1 to MEDIUM for the B-bra identi-
ified by the sorting instructions.

Get back in the loop for the completion
test and address modification.

Problem 3.2.

CLA TOTAL
STR TEMP
CLA ZRO
STR LOYAL
CHECK CLA WORKER+2
SUB K20
TRZ TWENTY
OVER CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
CLA CHECK
ADD K5
STR CHECK
TRU CHECK
TWENTY CLA LOYAL
ADD ONE
STR LOYAL
TRU OVER
STOP HLT
Problem 3.3.

CLA LIST
STR MEN
CLA ZRO
STR PASSES

To pick out the 0's, 1's, and 2's, substitute 3 to make them negative
(-3, -2, and -1).

CLA EM+3
SUB THREE
TRN DETAIL

END
CLA MEN
SUB ONE
STR MEN
TRZ STOP

CLA CYCLE
ADD FIVE
STR CYCLE

TRU CYCLE

CLA PASSES
ADD ONE
STR PASSES

Notice that counting occurs before the test for completion, even though
the instructions are written after; the computer skips directly down on
TRN DETAIL when a negative number appears in the accumulator. This is why
you must TRU back up after counting.

TRU END

STOP HLT
Problem 3.4.

You are interested in all numbers greater than 2, which means you are not interested in the numbers 0, 1, and 2—or the numbers less than 3.

So, if you subtract 3, the numbers you are not interested in will be negative in the accumulator, and you can TRN directly to the test for completion.

Take the numbers 0 through 4, for example. You do not want to count 0's, 1's, and 2's, but you do want to count 3's and 4's. See what happens when you subtract 3 from each.

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

do not count these

do count these

The complete program, therefore, is:

CLA FIGHT
TRZ STOP
STR FIGURE
CLA ZRO
STR LST

MORE CLA TRAIN+2
SUB CON3
TRN DONE
CLA LST
ADD CON1
STR LST

DONE CLA FIGURE
SUB CON1
STR FIGURE
TRZ STOP
CLA MORE
ADD CON4
STR MORE
TRU MORE

STOP HLT

Subtracting 3 makes the numbers you are not interested in negative, and you can TRN directly to the test for completion.

Numbers that are left are the ones to be counted.

The test for completion.

Address modification.
Problem 3.5.

First, look just at the sorting instructions. You want to count the 1's starting at symbolic location MONTH1 (for January) and the 2's starting at symbolic location MONTH2 (for February).

Thus, if you subtract 2, the 2's will be 0's in the accumulator and the 1's will be negative, or -1, in the accumulator. The sorting instructions are, therefore:

\[
\text{COMPUT CLA VACUUM+4 SUB TWO TRN MONTH1 TRZ MONTH2}
\]

The TRN and TRZ instructions can be reversed, of course. The program will still work the same.

The complete program would be as follows:

\[
\begin{align*}
\text{CLA TUBES} \\
\text{TRZ STOP} \\
\text{STR HOLD} \\
\text{CLA ZRO} \\
\text{STR JAN} \\
\text{STR FEB} \\
\text{COMPUT CLA VACUUM+4 SUB TWO TRN MONTH1 TRZ MONTH2} \\
\text{CLLZ HOLD} \\
\text{SUB ONE} \\
\text{STR HOLD} \\
\text{TRZ STOP} \\
\text{CLA COMPUT} \\
\text{ADD SIX} \\
\text{STR COMPUT} \\
\text{TRU COMPUT} \\
\text{MONTH1 CLA JAN ADD ONE} \\
\text{STR JAN TRU LAST} \\
\text{MONTH2 CLA FEB ADD ONE} \\
\text{STR FEB TRU LAST} \\
\text{STOP HALT}
\end{align*}
\]

These two blocks of instructions could be switched. Back to the test for completion.
Problem 3.6.

Of the numbers 2, 4, and 6, the ones you want to count are 2 and 6. If you subtract 4, they become -2, 0, and +2. This allows you to TRN to count small-sized shirts, TRZ directly to the test for completion (these are medium-sized shirts and you're not interested), and count large-sized shirts directly after sorting.

CLA TOTAL
TRZ STOP
STR NUMBER
CLA ZRO
STR SMALL
STR LARGE

SORT
CLA SHIRT
SUB KON4
TRN COUNT
TRZ END
CLA LARGE
ADD KON1
STR LARGE

END
CLA NUMBER
SUB KON1
STR NUMBER
TRZ STOP
CLA SORT
ADD KON1
STR SORT
TRU SORT
COUNT
CLA SMALL
ADD KON1
STR SMALL
TRU END

STOP

2's are now -2
0's indicate medium-sized shirts
Whatever is left started out as a 6
and is still positive; count them here.

Don't forget to finish out the loop.
Problem 3.7.

You want to subtract 8 to make the "middle" number zero. Then you can TRZ to count marginal men, TRN to count the men who flunked, and count the men accepted after the rest have been eliminated. Since the largest number available for subtraction is a 4, simply subtract it twice.

```
CLA PUPIL
STR MEN
CLA XOR
STR FLUNK
STR MARGIN
STR ACCEPT

LOOPER  CLA OFFCANT
        SUB FOUR
        SUB FOUR
        TRN OUT
        TRZ MAYBE
        CLA ACCEPT
        ADD ONE
        STR ACCEPT

LAST   CLA MEN
        SUB ONE
        STR MEN
        TRZ STOP
        CLA LOOPER
        ADD FOUR
        STR LOOPER
        TRU LOOPER

OUT    CLA FLUNK
        ADD ONE
        STR FLUNK
        TRU LAST

MAYBE  CLA MARGIN
        ADD ONE
        STR MARGIN
        TRU LAST

STOP   HLY
```

Now you have three answer locations.
Zero them all.

This marks the end of the loop in terms of the actual order in which instructions are carried out. The counting instructions are written at the end, but they are performed before the test for completion and address modification.
Problem 3.8.

The numbers 0 through 6 represent tubes that are all right. Tubes represented by 7 through 12 are to be checked, and numbers 13 and greater are for tubes to be replaced.

If you subtract 7, the numbers 0-6 will be negative and you can TRN directly to the test for completion. The numbers 7-12, the tubes to be checked, will become numbers 0-5. Therefore, if you then subtract again, using a 6, the tubes to be checked will be represented by negative numbers and you can TRN to count them. Anything left is a tube to be replaced.

There is another way to solve this problem, perhaps the one you chose. If you start by subtracting 6, the tubes that are all right will be represented by either negative or zero numbers, and you can TRN and TRZ to the test for completion. If you then subtract another 6, the tubes to be checked have negative or zero numbers, and you can TRN and TRZ to count them.

Assuming you got the rest of the program correct, the sorting instructions will be either of the following:

```
CLA TUBE
SUB SIX
SUB ONE
TRN TEST
SUB SIX
TRN LOOK
CLA REMOVE
ADD ONE
STR REMOVE
```

```
CLA TUBE
SUB SIX
TRN TEST
TRZ TEST
SUB SIX
TRN LOOK
CLA REMOVE
ADD ONE
STR REMOVE
```
### Problem 4.1.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST CLA COST</td>
<td>Change COST to COST+1.</td>
</tr>
<tr>
<td>SECOND STR COST+100</td>
<td>Change COST+100 to COST+101.</td>
</tr>
<tr>
<td>CLA K100</td>
<td>On the second loop, COST+1 will go into COST+101.</td>
</tr>
<tr>
<td>SUB K1</td>
<td></td>
</tr>
<tr>
<td>STR K100</td>
<td></td>
</tr>
<tr>
<td>TRZ STOP</td>
<td></td>
</tr>
<tr>
<td>CLA FIRST</td>
<td></td>
</tr>
<tr>
<td>ADD K1</td>
<td></td>
</tr>
<tr>
<td>STR FIRST</td>
<td></td>
</tr>
<tr>
<td>CLA SECOND</td>
<td></td>
</tr>
<tr>
<td>ADD K1</td>
<td></td>
</tr>
<tr>
<td>STR SECOND</td>
<td></td>
</tr>
<tr>
<td>TRU FIRST</td>
<td></td>
</tr>
<tr>
<td>STOP HLT</td>
<td></td>
</tr>
</tbody>
</table>

### Problem 4.2.

This isn't as tough as it looks. If you gave up on it, read the clues below and try it again before looking at the answer.

The relocation instructions are:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA COST</td>
<td></td>
</tr>
<tr>
<td>STR VALUE</td>
<td></td>
</tr>
<tr>
<td>CLA ZRO</td>
<td></td>
</tr>
<tr>
<td>STR VALUE+1</td>
<td></td>
</tr>
</tbody>
</table>

On the next loop, you want to relocate COST+1 into VALUE+2 and put zero into VALUE+3. So all three addresses must be changed during address modification. Ask yourself what addresses you need in place of COST, VALUE, and VALUE+1 for the second loop.

Now try the problem again and then turn to the next page for the answer.
Problem 4.2. (continued)

<table>
<thead>
<tr>
<th>FIRST</th>
<th>CLA COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOND</td>
<td>STR VALUE</td>
</tr>
<tr>
<td>THIRD</td>
<td>STR VALUE+1</td>
</tr>
<tr>
<td></td>
<td>CLA CON100</td>
</tr>
<tr>
<td></td>
<td>SUB CON1</td>
</tr>
<tr>
<td></td>
<td>STR CON100</td>
</tr>
<tr>
<td></td>
<td>TRZ STOP</td>
</tr>
<tr>
<td></td>
<td>CLA FIRST</td>
</tr>
<tr>
<td></td>
<td>ADD CON1</td>
</tr>
<tr>
<td></td>
<td>STR FIRST</td>
</tr>
<tr>
<td></td>
<td>CLA SECOND</td>
</tr>
<tr>
<td></td>
<td>ADD CON2</td>
</tr>
<tr>
<td></td>
<td>STR SECOND</td>
</tr>
<tr>
<td></td>
<td>CLA THIRD</td>
</tr>
<tr>
<td></td>
<td>ADD CON2</td>
</tr>
<tr>
<td></td>
<td>STR THIRD</td>
</tr>
<tr>
<td></td>
<td>TRU FIRST</td>
</tr>
<tr>
<td>STOP</td>
<td>HLT</td>
</tr>
</tbody>
</table>

The address (the name) of the location is the same, but its contents are now different.

Add a 1, not a 2.
Problem 4.3

CLA VOLUMS
STR CHECK
CLA ZRO
STR OUT

READER CLA BOOK
TRZ DONE

MARK STR FINE
CLA OUT
ADD K1
STR OUT

DONE CLA CHECK
SUB K1
STR CHECK
TRZ STOP

CLA READER
ADD K1
STR READER
CLA MARK
ADD K1
STR MARK

STOP TRU READER

CLA TOTAL
STR COUNT
FIRST CLA TUBE+2
SECOND STR TUBE+1
CLA COUNT
SUB ONE
STR COUNT
TRZ STOP
CLA FIRST
ADD TWO
STR FIRST
CLA SECOND
ADD ONE
STR SECOND

STOP TRU FIRST

CLA MARK
ADD K1
STR MARK

STOP TRU READER

Problem 4.4.

Here's what you want: TUBE TUBE+1 TUBE+2
And here's how to do it:

CLA TOTAL
STR COUNT
FIRST CLA TUBE+2
SECOND STR TUBE+1
CLA COUNT
SUB ONE
STR COUNT
TRZ STOP
CLA FIRST
ADD TWO
STR FIRST
CLA SECOND
ADD ONE
STR SECOND

STOP TRU FIRST

CLA MARK
ADD K1
STR MARK

STOP TRU READER

Modify BOOK to BOOK+1.

Modify FINE to FINE+1.

If BOOK has a 1, it is still in the accumulator and can be stored, or relocated, in FINE.

The 1 cannot be added to OUT with another STR command, however, since STR first erases the location addressed, before copying in the number from the accumulator.
Problem 4.5.

Look at this problem in terms of a block flow chart. The sorting instructions subtract 2 (the "middle" number), making the numbers in the accumulator negative, zero, or positive. Negative numbers then indicate company grade officers, zeroes are field grade, and positive numbers are for general officers, which can then be eliminated by going directly to the test for completion.

Notice in the complete program below that two words are used, but one of them appears twice (in the two counting blocks) which means that it must be modified in both places, resulting in three blocks of address modification instructions.

![Flow Chart Diagram]

- **CLA TOTAL**
- **STR HOLD**
- **CLA ZERO**
- **STR FINE**
- **STR WARN**
- **TICKET**
- **CLA DRIVER**
- **SUB K2**
- **TRN BAR**
- **TRZ LEAF**
- **OVER**
- **CLA HOLD**
- **SUB K1**
- **STR HOLD**
- **TRZ STOP**
- **CLA TICKET**
- **ADD K3**
- **STR TICKET**
- **CLA BAR**
- **ADD K3**
- **STR BAR**

(continued in next column)
Problem 5.1.

You have only one symbolic location (besides STOP). Therefore, you can't start the loop with CLA ZRO since you need the symbolic location to modify TRASH. The solution is to CLA ZRO as the last instruction in program preparation and also as the last instruction in the loop. Remember that transfer commands do not change numbers in the accumulator, so it is still there for the STR TRASH instruction.

CLA ITEM
STR SAVE (81 is obviously greater than zero; there is no need to insert a TRZ STOP instruction.)
CLA ZRO

NEXT STR TRASH
CLA SAVE
SUB KON
STR SAVE
TRZ STOP
CLA NEXT
ADD KON
STR NEXT
CLA ZRO This is the critical instruction.
TRU NEXT

STOP HLT

Incidentally, you may have noticed that just about any problem can be solved in several different ways. Take the adding instructions you learned long ago.

To add COST repeatedly into TOTAL: CLA COST
ADD TOTAL
STR TOTAL

But you could just as easily write: CLA TOTAL
ADD COST
STR TOTAL

The instructions need not occur as a single block of consecutive instructions either. You add COST into TOTAL in the following ways:

CLA ZRO -- or -- CLA ZRO
STR TOTAL
REPEAT -- or -- REPEAT
ADD TOTAL
STR TOTAL
test for
 completion
address
modification
CLA TOTAL
ADD COST
TRU REPEAT
STOP HLT

STOP HLT
Problem 5.2.

- CLA NUM
- SUB ONE
- STR HOLD
- FIRST CLA MAN
- ADD ONE
- TOTAL STR TEM
- CLA HOLD
- SUB ONE
- STR HOLD
- TRZ STOP
- CLA FIRST
- ADD ONE
- STR FIRST
- CLA TOTAL
- ADD ONE
- STR TOTAL
- TRU FIRST
- STOP HLT

Problem 5.3.

- CLA ZRO
- STR ANSWER
- ITEM CLA SORT
- TRZ LAST
- TRN LAST
- CLA ANSWER
- ADD KON
- STR ANSWER
- LAST CLA TOTAL
- SUB KON
- STR TOTAL
- TRZ STOP
- CLA ITEM
- ADD KON
- STR ITEM
- TRU ITEM
- STOP HLT

Problem 5.4.

- CLA ITEMS
- TRZ STOP
- STR TEM
- CLA ZRO
- STR NEED
- STR EXCESS
- COMPUT CLA SUPPLY +2
- SUB KON2
- TRN ORDER
- TRZ DONE
- CLA EXCESS
- ADD KON1
- STR EXCESS
- DONE CLA TEMP
- SUB KON1
- STR TEMP
- TRZ STOP
- CLA COMPUT
- ADD KON6
- STR COMPUT
- TRU COMPUT
- ORDER CLA NEED
- ADD KON1
- STR NEED
- TRU DONE
- STOP HLT
Problem 5.5.

This problem asks you to count the 20's (persons who will be 21 next year) and the numbers greater than 20 (persons 21 or older now). So if you subtract 20, negative numbers can be eliminated, zeroes are persons 20 years old, and what's left are persons 21 and older.

CLA ZRO
STR NOW
STR LATER

COMPUT CLA DATA
SUB CONST
ADD CON1
TRN TEST
TRZ NEXTYR

CLA NOW
ADD CON1
STR NOW

TEST CLA TOWN
SUB CON1
STR TOWN
TRZ STOP

CLA COMPUT
ADD CON1
STR COMPUT
TRU COMPUT

NEXTYR CLA LATER
ADD CON1
STR LATER
TRU TEST

STOP HLT

To subtract 20, subtract 21 and add 1.

This problem could be solved a variety of ways. This is one of the shorter solutions, but yours may work just as well. The important criterion is not how long a program is, but whether it works.
Problem 5.6.

CLA OFFC AN
TRZ STOP
STR MEN
CLA ZRO
STR PLUS
STR VALUE

DATA
CLA APPLY+4
SUB KON2
TRN ACCEPT

LAST
CLA MEN
SUB KON1
STR MEN
TRZ STOP
CLA DATA
ADD KON5
STR DATA
CLA ACCEPT
ADD KON5
STR ACCEPT
TRU DATA

ACCEPT
CLA APPLY+3
ADD VALUE
STR VALUE
CLA PLUS
ADD KON1
STR PLUS
TRU LAST

STOP
HLT

1 (for acceptance) - 2 = -1

Get the address for the next candidate's test score, whether or not it is actually used.

When these instructions are used, they are performed before the test for completion, even though they are written after.

(If test scores are added before accepted candidates are counted, only one symbolic location is needed.)
ANSWERS TO PROBLEMS OF PHASE IV: ADVANCED TECHNIQUES

Problem 1.1.

CLA STOCK1
STR TEMP
CLA STOCK2
STR TEMP+1
CLA ZRO
STR VALSTK
STR VALSTK+1
STR TOTAL
CLA VALSTK
ADD VALUE
STR VALSTK
CLA TEMP
SUB KON
STR TEMP
TRZ *+2
TRU *-7
CLA VALSTK+1
ADD VALUE+1
CLA TEMP+1
SUB KON
STR TEMP+1
TRZ *+2
TRU *-7
CLA VALSTK
ADD VALSTK+1
STR TOTAL
HLT

Problem 1.2.

CLA STOCK
STR TEMP
CLA ZRO
STR TOTAL
CLA TUBE
TRZ COMPUT
CLA VALUE
ADD TOTAL
STR TOTAL
CLA VALSTK
LAST
CLA TEMP
SUB KON
STR TEMP
TRZ *+2
ADD KON
STR NEXT
TRU NEXT
COMPUT
CLA VALUE+1
ADD TOTAL
STR TOTAL
TRU LAST
HLT
Problem 1.3.

CLM NUM
TRZ STOP
STR COUNT
CLA ZRO
STR HQDT
STR MS
STR NOTREQ
CLA INFO+5
SUB K2
TRN HQQTOT
TRZ MASEC
CLA NOTREQ
ADD K1
STR NOTREQ
CLA COUNT
SUB K1
STR COUNT
TRZ STOP
CLA LAST-7
ADD K6
STR LAST-7
TRU LAST-7
HQQTOT
CLA HQ DT
ADD K1
STR HQ DT
TRU LAST
MASEC
CLA MS
ADD K1
STR MS
TRU LAST
STOP HLT

Problem 1.4.

CLA M ME
TRZ STOP
CLA ZRO
STR TOTEM
DATA CLA INFO
TRN EN
CLA M M
SUB K1
STR M ME
STR M M
CLA DATA
ADD K3
STR DATA
TRU DATA
CLA LAST-i STOP HLT
CLA M ME
ADD K1
STR TOTEM
ADD K1
STR M M
TRU DATA+2
CLA HDQTS ADD K1
CLA HDQTS
CLA HDQTS
CLA HDQTS
CLA HDQTS
CLA HDQTS
CLA HDQTS
CLA HDQTS
CLA HDQTS
<table>
<thead>
<tr>
<th>Problem 1.5.</th>
<th>Problem 1.6.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA PERSON</td>
<td>CLA COUNT</td>
</tr>
<tr>
<td>STR HOLD</td>
<td>TRZ STOP</td>
</tr>
<tr>
<td>CLA ZRO</td>
<td>CLA ZRO</td>
</tr>
<tr>
<td>STR UP</td>
<td>STR TOT</td>
</tr>
<tr>
<td>RANK</td>
<td>NEXTOT</td>
</tr>
<tr>
<td>CLA RATING</td>
<td>CLA STAT</td>
</tr>
<tr>
<td>SUB KON</td>
<td>TRN AGAIN</td>
</tr>
<tr>
<td>TRZ SECOND</td>
<td>CLA COUNT</td>
</tr>
<tr>
<td>LAST</td>
<td>SUB K1</td>
</tr>
<tr>
<td>CLA HOLD</td>
<td>STR COUNT</td>
</tr>
<tr>
<td>SUB KON</td>
<td>TRZ STOP</td>
</tr>
<tr>
<td>STR HOLD</td>
<td>CLA NEXTOT</td>
</tr>
<tr>
<td>TRZ STOP</td>
<td>ADD K1</td>
</tr>
<tr>
<td>CLA RANK</td>
<td>STR NEXTOT</td>
</tr>
<tr>
<td>ADD K5</td>
<td>TRU NEXTOT</td>
</tr>
<tr>
<td>STR RANK</td>
<td>AGAIN</td>
</tr>
<tr>
<td>TRU RANK</td>
<td>CLA TOT</td>
</tr>
<tr>
<td>SECOND</td>
<td>ADD K1</td>
</tr>
<tr>
<td>CLA RATING+2</td>
<td>STR TOT</td>
</tr>
<tr>
<td>SUB TIME</td>
<td>CLA AGAIN</td>
</tr>
<tr>
<td>ADD KON</td>
<td>ADD K1</td>
</tr>
<tr>
<td>TRN LAST</td>
<td>STR AGAIN</td>
</tr>
<tr>
<td>CLA UP</td>
<td>TRU NEXTOT+2</td>
</tr>
<tr>
<td>ADD KON</td>
<td>STOP</td>
</tr>
<tr>
<td>STR UP</td>
<td>HLT</td>
</tr>
<tr>
<td>TRU LAST</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td></td>
</tr>
<tr>
<td>HLT</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1.7a

CLA COMPY OLD
STR HOLD CLA RECORD STR UPDATE
CLA RECORD+1 STR UPDATE+1
CLA RECORD+2 STR UPDATE+2
CLA RECORD+3 STR UPDATE+3
CLA HOLD
SUB K1
STR HOLD
TRZ STOP
CLA OLD
ADD K3
STR OLD
CLA OLD+1
ADD K4
CLA OLD+2
ADD K3
CLA OLD+3
ADD K4
CLA OLD+4
ADD K3
CLA OLD+5
ADD K4
CLA OLD+6
ADD K4
CLA OLD+7
ADD K4
CLA OLD+8
ADD K4
TRU OLD
STOP HLT

Problem 1.7b

CLA COMPY
STR NUM
CLA ZRO
STR TOTAL
CLA UPDATE+3
ADD TOTAL
STR NUM
CLA NUM
SUB K1
STR NUM
CLA NUM
TRZ STOP
CLA SUM
ADD K4
STR SUM
TRU SUM
STOP HLT

Problem 2.1.

SUB COST, IR3

Problem 2.2.

STR BOOK, IR1

Problem 2.3.

CLA IR1
ADD ONE
STR IR1
Problem 2.4.

CLA MEN
STR TEMP
CLA ZRO
STR I2
SELL ADD SALES, I2
STR TOTAL
CLA TEMP
SUB ONE
STR TEMP
TRZ STOP
CLA I2
ADD ONE
STR I2
CLA TOTAL
TRU SELL
STOP HLT

Problem 2.5.

CLA TYPES
STR POP
CLA ZRO
STR I1
HIT ADD RECORD, IR1
STR SALES
CLA POP
SUB K1
STR POP
TRZ STOP
CLA I1
ADD K1
STR I1
CLA SALES
TRU HIT
STOP HLT

Problem 2.6.

CLA HELP
STR I4
CLA ZRO
STR I3
TOTAL ADD RECORD+2, I3
STR REGPAY
CLA I4
SUB CON1
STR I4
TRZ STOP
CLA I3
ADD CON4
STR I3
CLA REGPAY
TRU TOTAL
STOP HLT

Problem 3.1.

LOD ZRO,, IR1

Problem 3.2.

LOD MAN,, IR2

Problem 3.3.

LOD COUNT,, IR3

Problem 3.4.

LOD ZRO,, IR2
LOD TOTAL,, IR1
CLA ZRO
Problem 3.5.

LOD ZRO,,IR3
LOD TEN,,IR1
CLA ZRO
CRAVAT ADD TIE,,IR3
STR ANSWER
CLA IR1
SUB ONE
STR IR1
TRZ STOP
CLA IR3
ADD ONE
STR IR3
CLA ANSWER
TRU CRAVAT
STOP HLT

Problem 3.6.

LOD ITEMS,,IR1
LOD ZRO,,IR2
CLA ZRO
REPEAT ADD TUBE+3,,IR2
STR HOURS
CLA IR1
TRZ STOP
SUB ONE
STR IR1
TRZ STOP
CLA IR2
ADD FOUR
STR IR2
CLA HOURS
TRU REPEAT
STOP HLT

Problem 4.1.

LOD MEN,,IR3
LOD ZRO,,IR1
CLA ZRO
ADD PUSH,,IR1
TRX *=1,IR1,1
STR BONUS
HLT

Problem 4.2.

LOD MEDICS,,IR3
LOD ZRO,,IR2
CLA ZRO
ADD DOCTOR,,IR2
TRX *=1,IR2,1
STR CALLS
HLT

Problem 4.3.

LOD ARRIVE,,IR2
LOD ZRO,,IR1
LOD LEAVE,,IR4
LOD ZRO,,IR3
CLA ZRO
ADD IN,,IR1
TRX *=1,IR1,1
ADD OUT,,IR3
TRX *=1,IR3,1
STR TOTAL
HLT
Problem 5.1.
LDX 6,IR1,0
CLA ZRO
ADD TYPE,IR1
TRX *=1,IR1,1
STR TOTAL
HLT

Problem 5.2.
LDX 4,IR2,0
CLA ZRO
ADD PAY,IR2
TRX *=1,IR2,1
STR GROSS
HLT

Problem 5.3.
LDX 3,IR2,0
CLA PAY
ADD PAY+1,IR2
TRX *=1,IR2,1
STR GROSS
HLT

Problem 6.1.
START CLA STOCK
STR TEMP
CLA ZRO
STR VALSTK
CLA VALSTK
ADD VALUE
STR VALSTK
DONE CLA TEMP
SUB ONE
STR TEMP
TRZ DONE+8
CLA DONE-2
ADD ONE
STR DONE-2
TRU DONE-3
HLT

Problem 6.2.
CLA STOCK
STR TEMP
CLA ZRO
STR VALSTK
COMPUT CLA VALSTK
ADD VALUE
STR VALSTK
CLA TEMP
SUB ONE
STR TEMP
TRZ *=5
CLA *=6
ADD ONE
STR *=8
TRU COMPUT
HLT

Problem 6.3.
ADD PAY,IR3

Problem 6.4.
STR COST,IR2

Problem 6.5.
LOD VALUE,,IR4
Problem 6.6.

```
LOD ZRO,IR1
LOD TOTUBE,,IR2
CLA ZRO
REPEAT ADD TUBE+1,IR1
STR VALUE
CLA IR2
TRZ STOP
SUB KON1
STR IR2
TRZ STOP
CLA IR1
ADD KON3
STR IR1
CLA VALUE
TRU REPEAT
STOP HLT
```

Problem 6.8.

```
LOD ZRO,IR2
LOD TUNE,,IR3
CLA ZRO
ADD TYPE,IR2
TRX *-1,IR2,1
STR ANSWER
HLT
```

Problem 6.9.

```
LOD ZRO,IR3
LOD RENT,,IR4
CLA ZRO
ADD INCOME,IR3
TRX *-1,IR3,1
STR ALL
HLT
```

Problem 6.7.

```
LOD ZRO,IR1
LOD DAY,IR2
CLA ZRO
AGAIN ADD INFO+3,IR1
STR TOTAL
CLA IR2
TR2 STOP
SUB ONE
STR IR2
TRZ STOP
CLA IR1
ADD FOUR
STR IR1
CLA TOTAL
TRU AGAIN
STOP HLT
```

Problem 6.10.

```
LDZ 70,IR4,0
CLA ZRO
ADD PAY,IR4
TRX*-1,IR4,1
STR TOTSAI
HLT
```

Problem 6.11.

```
LDX 20,IR1,0
CLA ZRO
ADD PRICE,IR1
TRX *-1,IR1,1
STR ALL
HLT
```
Problem 6.12.

LOD ZRO,,IR1
LOD STOCK,,IR2
CLA ZRO
STR TOT
CLA TUBE,IR1
SUB KON
TRX *+6
CLA VALUE
ADD TOT
STR TOT
TRX *-6,IR1,1
HLT
CLA VALUE+1
ADD TOT
STR TOT
TRU *-5

Problem 6.13.

LOD PERSON,,IR4
LOD ZRO,,IR3
CLA ZRO
STR UP
CLA RATING,IR3
SUB KON
TRZ *+6
ADD KON
TRX *-4,IR3,1
HLT
STR UP
TRU *-6
CLA RATING+2
SUB TIME
TRN *-9
TRZ *-10
TRU *-8


LDX 90,IR2,0
CLA ZRO
STR INV
CLA UNI,,IR2
TRZ *+6
ADD INV
ADD VALUE
STR INV
TRX *-5,IR2,1
HLT
CLA VALUE+1
ADD INV
STR INV
TRU *-5

Problem 6.15.

LDX 6,IR1,0
CLA ZRO
ADD TYPE,IR1
TRX *-1,IR1,1
STR TOTAL
LOD TOTAL,IR4
LOD ZRO,IR3
CLA ZRO
ADD COST,IR3
TRX *-1,IR3,1
STR VALUE
HLT