SMALLEST NUMBER IN AN ARRAY

EXP NO: 16

AIM: To find the smallest number from an array using 8085 processor.

ALGORITHM:

1) Load the address of the first element of the array in HL pair.
2) Move the count to B register.
3) Increment the pointer.
4) Get the first data in A register.
5) Decrement the count.
6) Increment the pointer.
7) Compare the content of memory addressed by HL pair with that of A register.
8) If carry=1, go to step 10 or if carry=0 go to step 9
9) Move the content of memory addressed by HL to A register.
10) Decrement the count.

PROGRAM:

```
LXI H,2050
MOV C,M
DCR C
INX H
MOV A,M
LOOP1: INX H
CMP M
JC LOOP
MOV A,M
LOOP: DCR C
JNZ LOOP1
STA 2058
HLT
```

INPUT:
OUTPUT:
RESULT: Thus the program was executed successfully using 8085 processor simulator.

ASCENDING ORDER

EXP NO: 17

AIM: To compute ascending order of an array using 8085 processor.

ALGORITHM:

1) Initialize HL pair as memory pointer.
2) Get the count at memory and load it into C register
3) Copy it in D register (for bubble sort (N-1)) times required).
4) Get the first value in A register.
5) Compare it with the value at next location.
6) If they are out of order, exchange the contents of A register and memory.
7) Decrement D register content by 1
8) Repeat step 5 and 7 till the value in D register become zero.
9) Decrement the C register content by 1.
10) Repeat steps 3 to 9 till the value in C register becomes zero.

PROGRAM:

```
LOOP: LXI H,3500
MVI D,00
MVI C,05
LOOP1: MOV A,M
INX H
CMP M
JC LOOP2
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
MVI D,01
LOOP2: DCR C
JNZ LOOP1
MOV A,D
RRC
```

```
    JC LOOP
    HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

DESCENDING ORDER

EXP NO: 18

AIM: To compute descending order of an array using 8085 processor.

ALGORITHM:

1) Initialize HL pair as memory pointer.
2) Get the count at memory and load it into C register
3) Copy it in D register (for bubble sort (N-1)) times required).
4) Get the first value in A register.
5) Compare it with the value at next location.
6) If they are out of order, exchange the contents of A register and memory.
7) Decrement D register content by 1
8) Repeat step 5 and 7 till the value in D register become zero.
9) Decrement the C register content by 1.
10) Repeat steps 3 to 9 till the value in C register becomes zero.

PROGRAM:

```
LOOP: LXI H,3500
MVI D,00
MVI C,05
LOOP1: MOV A,M
INX H
CMP M
JNC LOOP2
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
MVI D,01
LOOP2: DCR C
JNZ LOOP1
MOV A,D
```

```
    RRC
    JC LOOP
    HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

ADDITION OF N NUMBERS

EXP NO: 19

AIM: To compute addition of N numbers using 8085 processor.

ALGORITHM:

1) Load the base address of the array in HL register pair.
2) Load the memory with data to be added.
3) Take it as count.
4) Initialize the accumulator with 00.
5) Add content of accumulator with content of memory.
6) Decrement count.
7) Load count value to memory location.
8) Repeat step 5.
9) Check whether count has become 0.
10) Halt.

PROGRAM:

```
LXI H,8000
MOV C,M
MVI A,00
MOV B,A
LOOP: ADD C
JNC SKIP
INR B
SKIP: DCR C
JNZ LOOP
LXI H,8007
MOV M,A
INX H
MOV M,B
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

SWAPPING OF NUMBERS

EXP NO: 20

AIM: To compute swapping of numbers using 8085 processor.

ALGORITHM:

1) Load a 8-bit number from memory location into accumulator.
2) Move value of accumulator into register H.
3) Load a 8-bit number from next memory location into accumulator.
4) Move value of accumulator into register D.
5) Exchange both the registers pairs.
6) Halt

PROGRAM:

```
LDA 2001
MOV B,A
LDA 2002
MOV C,A
STA 2003
MOV A,B
STA 2004
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

SQUARE OF NUMBER

EXP NO: 21

AIM: To compute square of number using 8085 processor.

ALGORITHM:

1) Load the base address of the array in HL register pair.
2) Assign accumulator as 0.
3) Load the content of memory location specified into register.
4) Add content of memory location with accumulator and decrement register content by 01.
5) Check if register holds 00, if so store the value of accumulator in memory location.

PROGRAM:

```
LXI H,8000
XRA A
MOV B,M
LOOP: ADD M
DCR B
JNZ LOOP
STA 8001
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

ONEs AND TWOs COMPLEMENT

EXP NO: 22

AIM: To compute one's and two's complement using 8085 processor.

ALGORITHM:

1) Load the base address of the array in a register pair.
2) Move the data from memory location into accumulator.
3) Convert all ones into zeros and zeros into ones.
4) Add 01 to the accumulator content.
5) Store the results of one's and two's complement.

PROGRAM:

```
LDA 3000
CMA
STA 3001
ADI 01
STA 3002
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

ROTATE LEFT OPERATION

EXP NO: 23

AIM: To compute rotation of given data in left without carry using 8085 processor.

ALGORITHM:

1) Load the base address of the array in HL register pair.
2) Move the data from memory location into accumulator.
3) Shift left the accumulator content for four times.
4) Store the result in the specified location.

PROGRAM:

```
MVI A,02
RLC
RLC
RLC
RLC
STA 2000
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

ROTATE RIGHT OPERATION

EXP NO: 24

AIM: To compute rotation of given data in right without carry using 8085 processor.

ALGORITHM:

1) Load the base address of the array in HL register pair.
2) Move the data from memory location into accumulator.
3) Shift right the accumulator content for four times left.
4) Store the result in the specified location.

PROGRAM:

```
MVI A,03
RRC
RRC
RRC
RRC
STA 2000
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.

LOGICAL OPERATIONS

EXP NO: 25

AIM: To compute various logical operations using 8085 processor.

ALGORITHM:

1) Load data to accumulator.
2) Load another data in register
3) Perform logical operations like AND, OR and XOR (Use ANA, ORA, XRA) with the accumulator content.
4) Store the result in specified memory location.

PROGRAM:

AND OPERATION:

```
MVI A,06
MVI B,04
ANA B
STA 2500
HLT
```

OR OPERATION:

```
MVI A,07
MVI B,06
ORA B
STA 2000
HLT
```

XOR OPERATION:

```
MVI A,03
MVI B,04
XRA B
STA 2000
HLT
```

INPUT:

OUTPUT:

RESULT: Thus the program was executed successfully using 8085 processor simulator.