

COECC08-MICROPROCESSORS AND MICROCONTROLLERS

PRACTICAL FILE



MADE BY: AMOGH GARG

ROLL NUMBER:2020UCO1688

NETAJI SUBAS UNIVERSITY OF TECHNOLOGY, NEW DELHI

**INDEX**

1. Generate number of Fibonacci series.

2. Clear all flags without using any data transfer instruction

3. Program to search for a number in a list.

4. Program to sort a list.

5. Program to copy a list from one part of the memory to another.

6. Multiply two numbers using successive additions.

7. Program to calculate square root of a number.

8. Program to calculate factorial using recursion.

**PROGRAM-1**

**CODE: Generate number of Fibonacci series.**

LXI H,3050H  
MVI C,08  
MVI B,00  
MVI D,01  
MOV M, B  
INX H   
MOV M, D  
JUMP: MOV A, B  
ADD D  
MOV B, D  
MOV D, A  
INX H  
MOV M, A  
DCR C  
JNZ JUMP  
HLT

**PROGRAM-2**

**CODE: Clear all flags without using any data transfer instruction.**

START: NOP  
XRA A  
INR A  
STC  
CMC  
HLT

**PROGRAM-3**

**CODE: Program to search for a number in a list.**

LXI H,2050H; //Loads number to be found

MOV B, M; //Stores it in B

LXI H,2000H; //Goes to starting of the list

MVI C,0AH; //Counter for list traversal

MVI D,05H; //Stores number of elements in array

LOOP: MOV A, M;

CMP B;

JZ BREAK; //Jumps if number found

INX H; //If not jumps to next num

DCR D; //Decrements counter

JNZ LOOP; //Goes back to comparing

LXI H,2051H; //Loads memory where o/p is to be stored

MVI M,00H; //Stores zero if not found

HLT;

BREAK: LXI H,2051H;

MVI M,01H; //Stores 1 if found

HLT;

**PROGRAM-4**

**CODE: Program to sort a list.**

START: NOP;

LXI H, 2000H; //This location stores size of list

MVI D,00H;

MOV C, M;

DCR C;

INX H; //List begins at 2001

COMPARE: MOV A, M;

INX H;

CMP M;

JC NEXT; //If first num smaller proceed to next

JZ NEXT; //If equal proceed to next

MOV B, M; //If not then swap

MOV M, A;

DCX H;

MOV M, B;

INX H;

MVI D,01H;

NEXT: DCR C; //Decrement after every iteration

JNZ COMPARE;

MOV A, D;

CPI 01H;

JZ START;

HLT;

**PROGRAM-5**

**CODE: Program to copy a list from one part of the memory to another.**

START: NOP; //Moving five number from 2000H to 2040H

MVI C,05;

LXI H,2000H;

LXI D,2040H;

JUMP: MOV A, M;

STAX D ; //Stores contents of A in D

INX H; //Goes to next in original list

INX D; //Goes to next in where to store

DCR C; //Decrements counter

JNZ JUMP;

HLT;

**PROGRAM-6**

**CODE: Multiply two numbers using successive additions.**

START: LXI H,2000H; //Gets first number

MOV B, M; //B has first number

INX H; //Gets second number

MOV D, M; //D has second number

XRA A; //A=0

MVI C,00H; //This is to store carry

LOOP: ADD B;

JNC SKIP;

INR C; //If carry increment c

SKIP: DCR D;

JNZ LOOP;

LXI H,2050H;

MOV M,C; //Store carry

INX H;

MOV M, A; //Store num

HLT;

**PROGRAM-7**

**CODE: Program to calculate square root of a number.**

START: NOP;

MVI D,01;

MVI E,01;

LDA 2050H;

LABEL: SUB D;

JZ JUMP;

INR D;

INR D;

INR E;

JMP LABEL;

JUMP: MOVA, E;

STA 2000H;

HLT;

**PROGRAM-8**

**CODE: Program to calculate factorial using recursion.**

START: LXI H,2000H;

MOV B, M;

MVI D, 01H;

LOOP: CALL LABEL

DCR B;

JNZ LOOP;

INX H;

MOV M, D;

HLT;

LABEL: MOV E, B;

MVI A, 00H;

LABEL2: ADD D;

DCR E;

JNZ LABEL2;

MOV D, A;

RET;