

8085 Opcode Sheet

	B	D	H	SP
LXI	1	11	21	31
INX	3	13	23	33
DCX	0B	1B	2B	3B
DAD	9	19	29	39
LDAX	0A	1A		
STAX	2	12		PSW
PUSH	C5	D5	E5	F5
POP	C1	D1	E1	F1

		C	NC	Z	NZ
JMP	C3	DA	D2	CA	C2
CALL	CD	DC	D4	CC	C4
RET	C9	D8	D0	C8	C0

	P	M	PE	PO
JMP	F2	FA	EA	E2
CALL	F4	FC	EC	E4
RET	F0	F8	E8	E0

MOV	B	C	D	E	H	L	M	A
B	40	41	42	43	44	45	46	47
C	48	49	4A	4B	4C	4D	4E	4F
D	50	51	52	53	54	55	56	57
E	58	59	5A	5B	5C	5D	5E	5F
H	60	61	62	63	64	65	66	67
L	68	69	6A	6B	6C	6D	6E	6F
M	70	71	72	73	74	75	76	77
A	78	79	7A	7B	7C	7D	7E	7F

SHLD	22
LHLD	2A
XCHG	EB
SPHL	F9
XTHL	E3
LDA	3A
STA	32
PCHL	E9

EI	FB	RRC	0F	CMA	2F		
DI	F3	RLC	07	STC	37	RIM	20
NOP	00	RAR	1F	CMC	3F	SIM	30
HLT	76	RAL	17	DAA	27		

R			S			T	
0	1	2	3	4	5	6	7
C7	CF	D7	DF	E7	EF	F7	FF

IN	DB
OUT	D3

WITH ACCUMULATOR							
ADI	C6	SUI	D6	ANI	E6	ORI	F6
ACI	CF	SBI	DF	XRI	EE	CPI	FE

ADD	80	81	82	83	84	85	86	87
ADC	88	89	8A	8B	8C	8D	8E	8F
SUB	90	91	92	93	94	95	96	97
SBB	98	99	9A	9B	9C	9D	9E	9F
ANA	A0	A1	A2	A3	A4	A5	A6	A7
XRA	A8	A9	AA	AB	AC	AD	AE	AF
ORA	B0	B1	B2	B3	B4	B5	B6	B7
CMP	B8	B9	BA	BB	BC	BD	BE	BF
INR	04	0C	14	1C	24	2C	34	3C
DCR	05	0D	15	1D	25	2D	35	3D
MVI	06	0E	16	1E	26	2E	36	3E
	B	C	D	E	H	L	M	A

Practical 3:

Title: To perform encryption & decryption of a given message using Ceaser Cipher

Code:

```
98  √ #include <stdio.h>
99  #include <string.h>
100
101  √ int main(){
102      char str[20];
103      printf("Enter a message\n");
104      gets(str); // stores the whole string including space character
105
106  √  char *ptr = str; // *ptr stores addresses of str
107      | | | | | // *ptr ="Hello World"
108      int len = strlen(str);
109  √  while (*ptr != '\0') // runs until it encounters '\0'
110      {
111          *ptr += 1;
112          ptr++;
113      }
114      printf("The encrypted message is: %s\n", str);
115      char str2[len];
116  √  for (int i = 0; i < len; i++)
117      {
118          str2[i] = str[i]; // stores the encrypted message into new string
119      }
120
121  √  char *ptr2 = str2; // *ptr2 will work the same for str2
122      | | | | | // * ptr= *****
123  √  while (*ptr2 != '\0')
124      {
125          *ptr2 -= 1; // this time the message is converted back into original
126          *ptr2++;
127      }
128      printf("The decrypted message is: %s\n", str2);
129      return 0;
130  }
```

Output:

```
Enter a message
Hello World
The encrypted message is: Ifmmp!Xpsme
The decrypted message is: Hello World
```

Practical 05

Aim: Write a program to find GCD of two numbers using Euclidean Algorithm.

Code:

```
243  #include<stdio.h>
244
245  // DM Practical 04:
246  // WAP to find the GCD(Greater common divisor) of a number
247  int main()
248  {
249      int x, y;
250      printf("Enter two numbers: \n");
251      printf("Enter 1st number: \n");
252      scanf("%d",&x);
253      printf("Enter 2nd number: \n");
254      scanf("%d",&y);
255
256      printf("The GCD of (x,y) is: ",x,y);
257      while(x!=0)
258      {
259          x=x%y;
260          if(y!=0)
261          {
262              y=x;
263              printf("%d\n",y);
264          }
265      }
266      return 0;
267  }
```

Output:

```
Enter two numbers:
Enter 1st number:
50
Enter 2nd number:
20
The GCD of (x,y) is: 10
```

```
Enter two numbers:
Enter 1st number:
108
Enter 2nd number:
72
The GCD of (x,y) is: 36
```