Assignment - 2

1. Write an assembly program to print the sum of the first “N” naturalnumbers.

Ex: N=5, Sum=1+2+3+4+5=15

```
MOV A,R4;
MOV B,A;
INC B;
MUL AB;
MOV B,#2;
DIV AB;
MOV R0,A;
END;
```

2. Write an assembly language program to check whether the given number isEVEN or ODD.

```
MOV A,R4;
ANL A,#01;
MOV R0,A;
END; 0 means even and 1 means odd
```

3. Implement an assembly language program to print the sum of the numbersgiven in an Array.

```
MOV R3,#0H;
MOV R0,#20H;
MOV A,@R0;
MOV R4,A;
MOV R0,#21H;
MOV A,#0H;
loop: ADD A,@R0;
JNC noCarry;
INC R3;
noCarry: INC R0;
DJNZ R4,loop;
MOV R0,#30H;
MOV @R0,A;
MOV R0,#31H;
MOV B,R3;
MOV @R0,B;
END;
```

4. Write an assembly language program to print the sum of all even numbersless than “N”, where “N” is a given positive number.

```
MOV R0,#20;
MOV B,@R0;
MOV R1,B;
MOV R5,#0H;
repeat: MOV A,R1;
ANL A,#01;
CJNE A, #00H,odd;
MOV A,R5;
ADD A,@R1;
MOV R5,A;
odd: DJNZ R1,repeat;
MOV A,R5;
MOV R1,A;
END;
```

5. Write an assembly language program for addition, subtraction and multiplication of two 16 bit numbers.

ADDITION :

```
MOV DPTR, #FFh ; Load the address of the first data into DPTR
MOVX A, @DPTR        ; Move the data from the address pointed by DPTR to accumulator A
INC DPTR             ; Increment DPTR to point to the next data
MOV R2, A            ; Move the data from accumulator A to register R2

MOV DPTR, #F0h ; Load the address of the second data into DPTR
MOVX A, @DPTR        ; Move the data from the address pointed by DPTR to accumulator A
ADD A, R2            ; Add the data in accumulator A with the data in register R2
MOV R3, A            ; Move the result to register R3

MOV A, R3            ; Move the result from register R3 to accumulator A
MOV DPTR, #2Ah ; Load the address of the result into DPTR
MOVX @DPTR, A        ; Move the result from accumulator A to the address pointed by DPTR
END
```

SUBTRACTION :

```
MOV DPTR, #FFh ; Load the address of the minuend into DPTR
MOVX A, @DPTR        ; Move the data from the address pointed by DPTR to accumulator A
MOV R2, A            ; Move the data from accumulator A to register R2

MOV DPTR, #F0h ; Load the address of the subtrahend into DPTR
MOVX A, @DPTR        ; Move the data from the address pointed by DPTR to accumulator A
CPL A                ; Take the one's complement of the subtrahend
INC A                ; Add 1 to get the two's complement
ADD A, R2            ; Add the two's complement of the subtrahend to the minuend
```

```
MOV R3, A            ; Move the result to register R3
MOV A, R3            ; Move the result from register R3 to accumulator A
MOV DPTR, #2Ah ; Load the address of the result into DPTR
MOVX @DPTR, A        ; Move the result from accumulator A to the address pointed by DPTR
END
```

MULTIPLICATION :

```
MOV DPTR, #FFh ; Load the address of the first data into DPTR
MOVX A, @DPTR ; Move the data from the address pointed by DPTR to accumulator A
MOV R2, A            ; Move the data from accumulator A to register R2
MOV DPTR, #02h  ; Load the address of the second data into DPTR MOVX A,
@DPTR  ; Move the data from the address pointed by DPTR to accumulator AMOV R3, A        ;
Move the data from accumulator A to register R3

MOV A, R2          ; Move the data from register R2 to accumulator A MUL AB
                   ; Multiply the data in accumulator A with the data in register B (R3)MOV
R4, A              ; Move the lower byte of the result to register R4
MOV R5, B          ; Move the upper byte of the result to register R5

MOV A, R4            ; Move the lower byte of the result from register R4 to accumulator A
MOV DPTR, #31h ; Load the address of the lower byte of the result into DPTR
MOVX @DPTR, A ; Move the lower byte of the result from accumulator A to the address pointed by DPTR

MOV A, R5            ; Move the upper byte of the result from register R5 to accumulator A
MOV DPTR, #32h ; Load the address of the upper byte of the result into DPTR
MOVX @DPTR, A ;Move the upper byte of the result from accumulator A to the address pointed by DPTR
END
```