

# Embedded Systems Lab

## Assignment 2

**Name – Neelangshu Roy**

**Reg No – 20214060**

**Group – C1**

1. Write an assembly program to print the sum of first “N” natural numbers. Ex:  
N=5, Sum=1+2+3+4+5=15

; 70h stores n, assume  $(n*(n+1))/2 < 256$

mov a, 70h

mov r0, a

mov r1, a

inc r1

mov b, #02h

anl a, #01h

jnz n\_odd

mov a, r0

div ab

mov b, r1

ljmp outer

n\_odd: mov a, r1

div ab

mov b, r0

outer: mul ab

mov 7fh, a ; 7fh stores  $(n*(n+1))/2$

end

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

; number n is in address 70h

mov 7fh, #00h

mov a, 70h ; copies value in grid box 70 to acc

anl a, #01h ; AND contents in acc with 01

; so if odd, last bit is set, else zero

jz next1

inc 7fh ; if odd, increase to 01

next1:

end

3. Program to transfer N = 05h bytes of data from the location A = 30h to location B = 40h.

; 70h stores no. of bytes to transfer

mov r0, #30h

mov r1, #40h

mov a, 70h

loop: mov r2, a

mov a, @r0

mov @r1, a

inc r0

inc r1

mov a, r2

dec a

jnz loop

end

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

; 70h stores n

mov r0, 70h

dec r0

mov a, r0

dec a

jz getout ; if n <= 2

mov a, r0

anl a, #01h

jz itsok

dec r0

itsok: mov r1, #00h

mov a, r0

loop: mov b, a

mov a, r1

add a, b

mov r1, a

mov a, b

sub a, #02h ; a-=2, a was already even

jnz loop

mov 7fh, r1 ; 7fh stores answer

getout:

end

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

Assumption – all results fit in 16-bit register pair, i.e., no carry or borrow exists.

; numbers are in 30h, 31h and 40h, 41h

; smaller 8 bits are in 31, 41h and larger ones in 30, 40h

Addition –

```
mov 50h, #00h
```

```
mov 51h, #00h
```

```
mov r0, #00h
```

```
mov a, 31h
```

```
mov r1, 41h
```

```
add a, r1
```

```
jnc next1
```

```
inc r0
```

```
next1: mov 51h, a
```

```
mov a, 30h
```

```
add a, r0
```

```
mov r0, 40h ; result's smaller 8 bits are in 51h
```

```
add a, r0 ; larger 8 bits in 50h
```

```
mov 50h, a
```

```
end
```

Subtraction –

```
mov a, 30h
```

```
mov r0, 40h
```

```
subb a, r0
```

```
mov r1, a
mov a, 31h
mov r0, 41h
subb a, r0
jnc next1
dec r1
next1: mov 51h, a
mov 50h, r1
end
```

Multiplication –

```
mov a, 31h
mov b, 41h
mul ab
mov 51h, a
mov r0, b
mov a, 30h
mov b, 40h
mul ab ; assume no carry is generated
add a, r0
mov 50h, a
end
```