**Embedded Systems Lab**

**Assignment 3**

**Name** – Neelangshu Roy

**Reg No** – 20214060

**Group** – C1

movc a, @a+pc

this takes the current value in pc, adds acc value to it,

and now this added value is actually the address of a cell

in the code memory and the value in that cell is now copied

into the acc.

Q1. Check if a number n is prime or not.

; 70h stores the value

; if prime 7fh is set to

; its smallest divisor,

; else it remains zero

mov 7fh, #00

mov a, 70h

subb a, #02

jc getout

mov r0, #02

loop: mov a, 70h

mov b, r0

div ab

mov a, #00

subb a, b

jz setit

inc r0

mov a, r0

subb a, 70h

jz getout

jmp loop

setit: mov 7fh, r0

getout:

end

Q2. Program for sum of array numbers.

; 70h stores length of array

; 71h stores address of arr[0]

; answer will be in 7eh(carry)

; and 7fh(remainder)

mov r0, 71h

mov a, 70h

mov r1, #00

mov r2, #00

loop: mov b, a

mov a, r1

add a, @r0

jnc next1

inc r2

next1:

mov r1, a

mov a, b

inc r0

dec a

jnz loop

mov 7eh, r2

mov 7fh, r1

end

Q3. Program to shift block of data from one location range to another even if ranges overlap.

; 70h stores length of segment

; 71h stores starting address of

; 1st segment

; 72h stores starting address of

; 2nd segment

mov a, 70h

mov r1, #7fh

mov r0, 71h

loop: mov r2, a

mov a, @r0

mov @r1, a

dec r1

inc r0

mov a, r2

dec a

jnz loop

mov a, 70h

mov r1, 72h

mov r0, #7fh

loop2: mov r2, a

mov a, @r0

mov @r1, a

inc r1

dec r0

mov a, r2

dec a

jnz loop2

end

Q4. Exchange data blocks of n bytes.

; 70h stores length of segment

; 71h stores starting address of

; 1st segment

; 72h stores starting address of

; 2nd segment

mov a, 70h

mov r1, #7fh

mov r0, 71h

loop: mov r2, a

mov a, @r0

mov @r1, a

dec r1

inc r0

mov a, r2

dec a

jnz loop

mov a, 70h

mov r1, 71h

mov r0, 72h

loop2: mov r2, a

mov a, @r0

mov @r1, a

inc r1

inc r0

mov a, r2

dec a

jnz loop2

mov a, 70h

mov r1, 72h

mov r0, #7fh

loop3: mov r2, a

mov a, @r0

mov @r1, a

inc r1

dec r0

mov a, r2

dec a

jnz loop3

end