Name : Sajid Islam

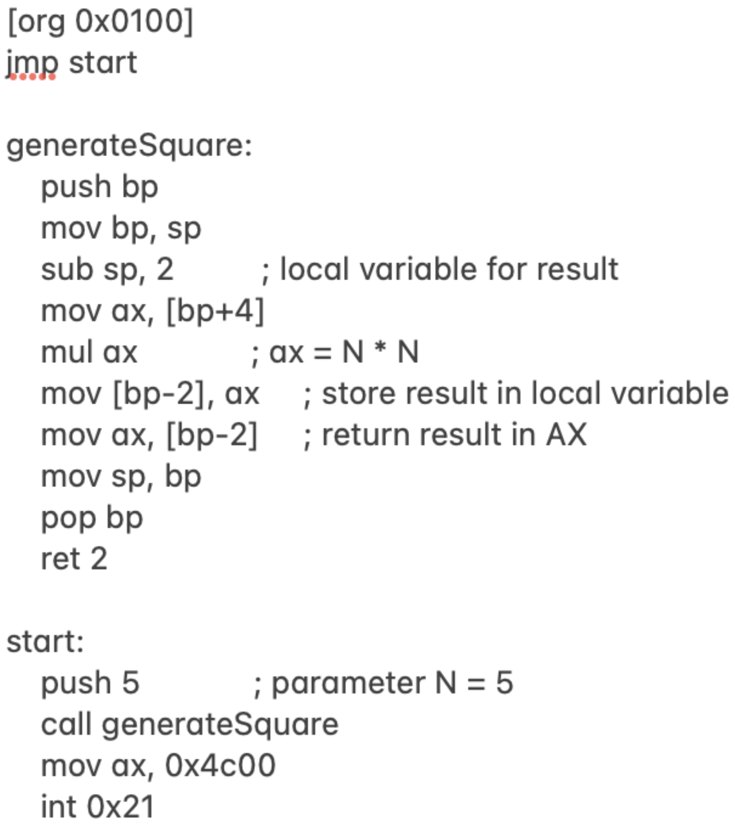
Instructor : Muhammad Hassaan

COAL

Assessment No 11

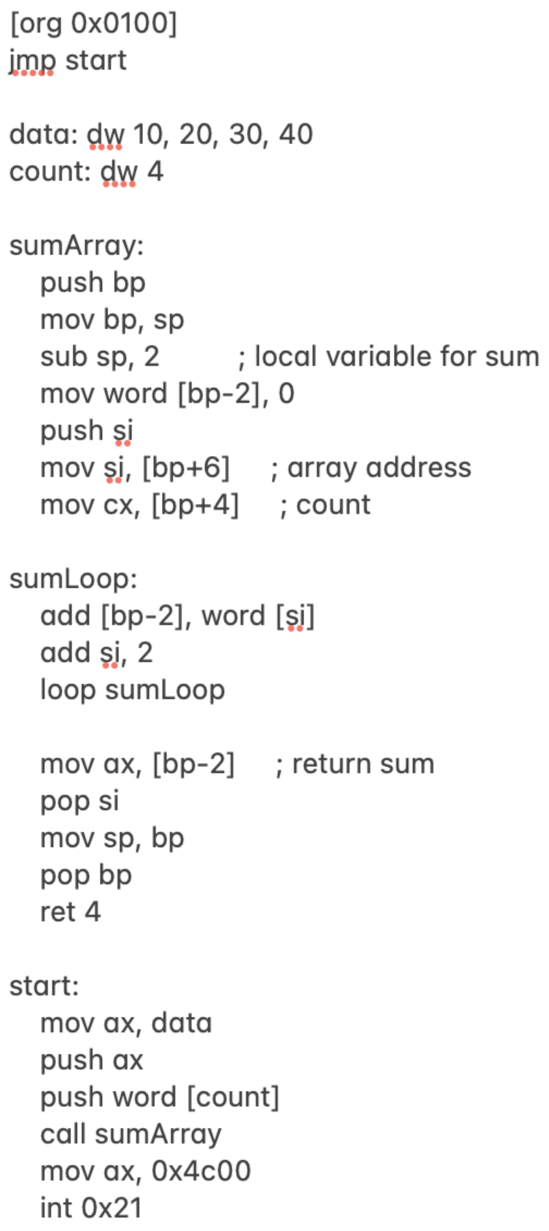
Roll No : 24p-0745

Q1



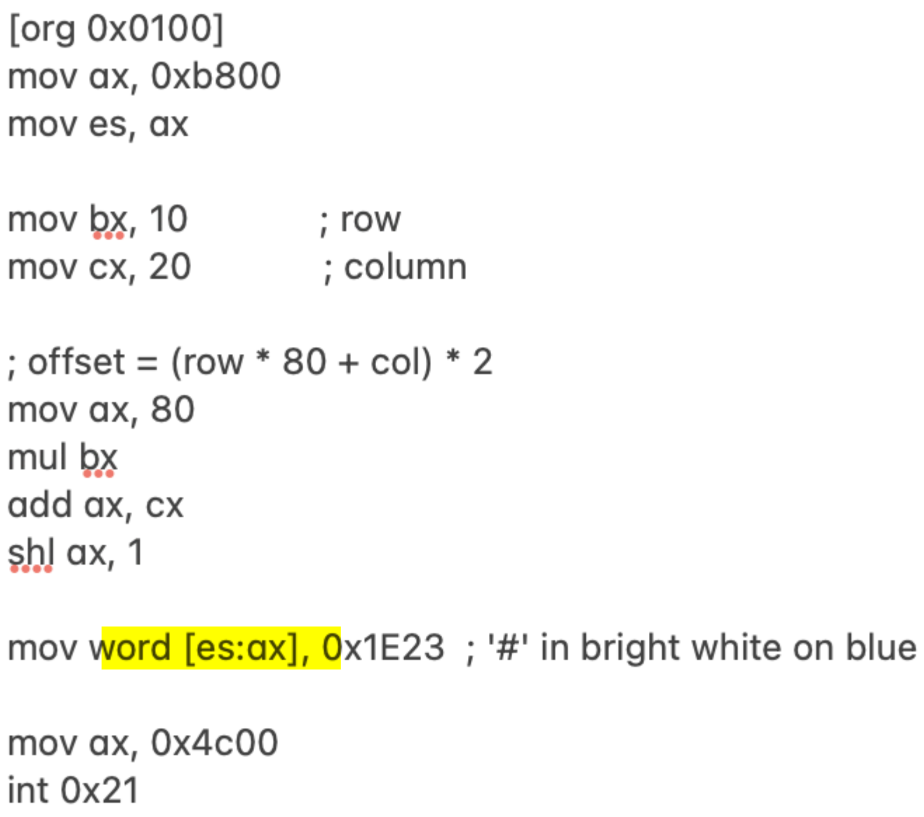
```
[org 0x0100]
jmp start

generateSquare:
   push bp
   mov bp, sp
   sub sp, 2          ; local variable for result
   mov ax, [bp+4]
   mul ax             ; ax = N * N
   mov [bp-2], ax     ; store result in local variable
   mov ax, [bp-2]     ; return result in AX
   mov sp, bp
   pop bp
   ret 2

start:
   push 5             ; parameter N = 5
   call generateSquare
   mov ax, 0x4c00
   int 0x21
```

Q2



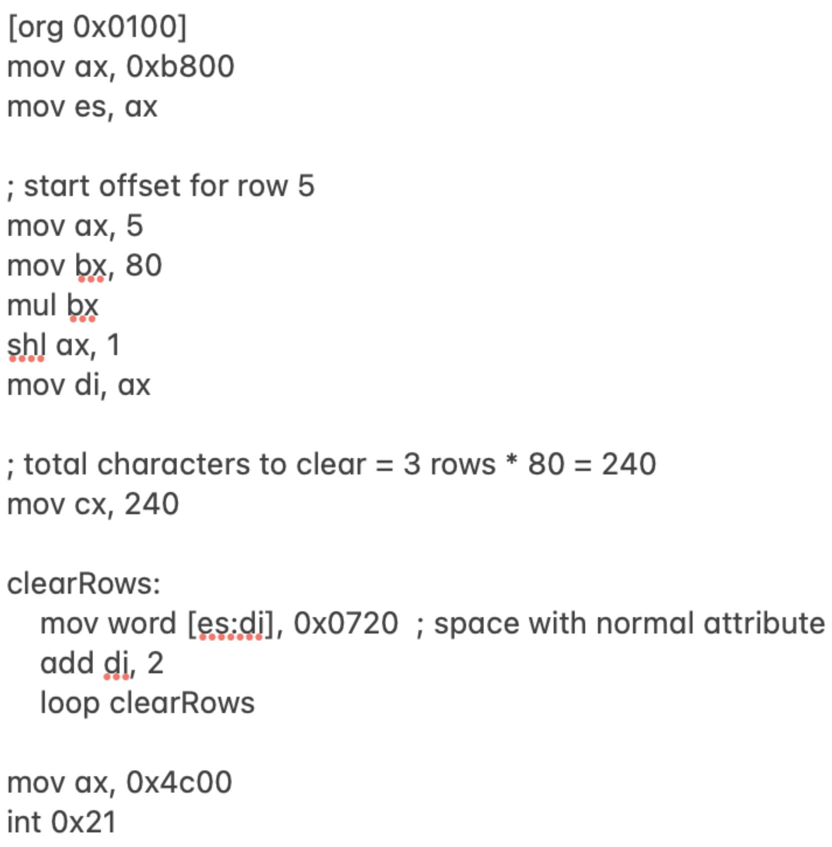
```
[org 0x0100]
jmp start

data: dw 10, 20, 30, 40
count: dw 4

sumArray:
   push bp
   mov bp, sp
   sub sp, 2          ; local variable for sum
   mov word [bp-2], 0
   push si
   mov si, [bp+6]     ; array address
   mov cx, [bp+4]     ; count

sumLoop:
   add [bp-2], word [si]
   add si, 2
   loop sumLoop

   mov ax, [bp-2]     ; return sum
   pop si
   mov sp, bp
   pop bp
   ret 4

start:
   mov ax, data
   push ax
   push word [count]
   call sumArray
   mov ax, 0x4c00
   int 0x21
```

Q3



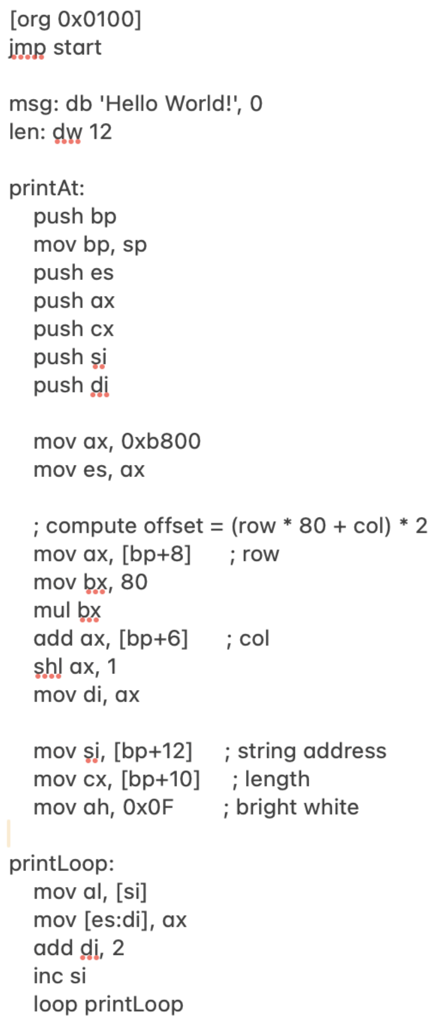
```
[org 0x0100]
mov ax, 0xb800
mov es, ax

mov bx, 10          ; row
mov cx, 20          ; column

; offset = (row * 80 + col) * 2
mov ax, 80
mul bx
add ax, cx
shl ax, 1

mov word [es:ax], 0x1E23  ; '#' in bright white on blue

mov ax, 0x4c00
int 0x21
```

Q4

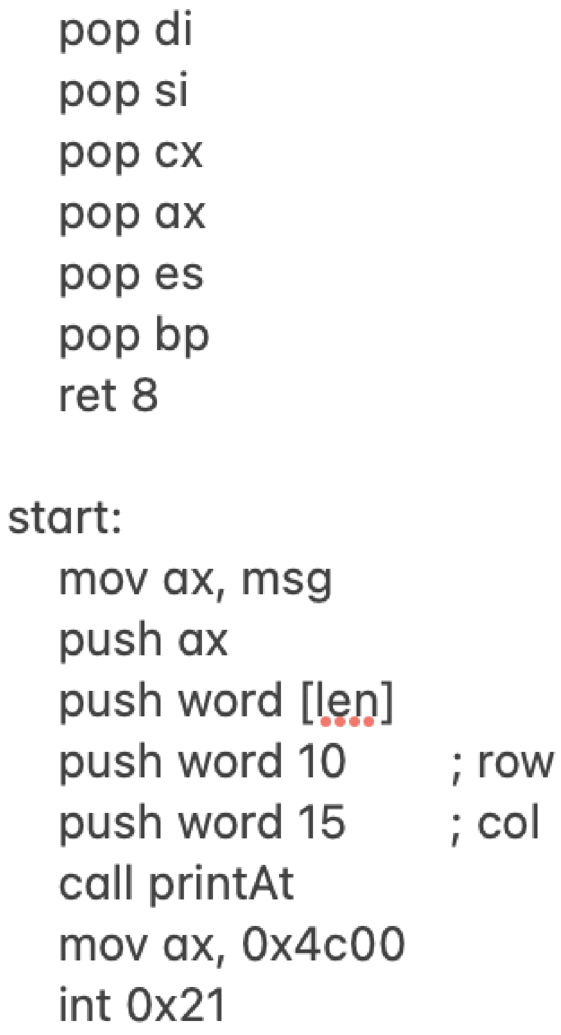


```
[org 0x0100]
mov ax, 0xb800
mov es, ax

; start offset for row 5
mov ax, 5
mov bx, 80
mul bx
shl ax, 1
mov di, ax

; total characters to clear = 3 rows * 80 = 240
mov cx, 240

clearRows:
   mov word [es:di], 0x0720  ; space with normal attribute
   add di, 2
   loop clearRows

mov ax, 0x4c00
int 0x21
```

Q5

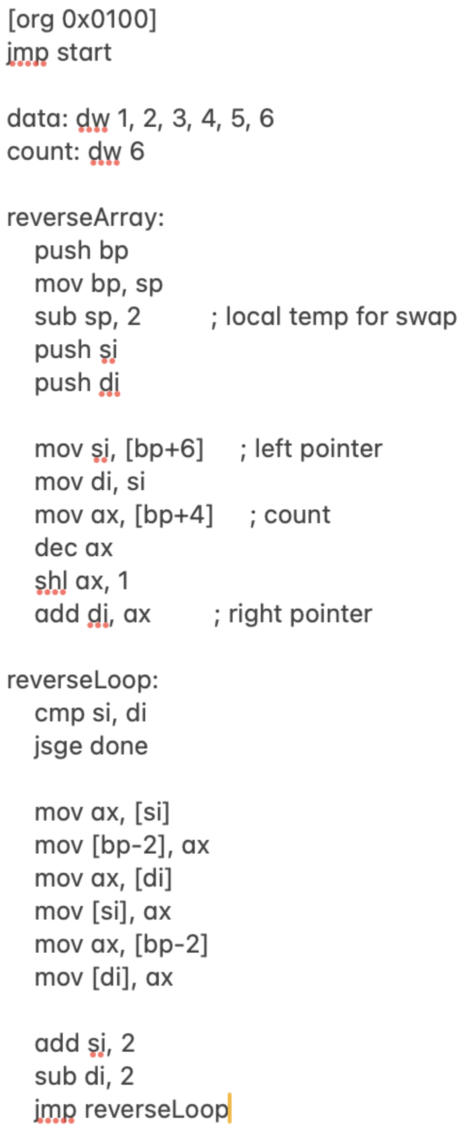


```
[org 0x0100]
jmp start

msg: db 'Hello World!', 0
len: dw 12

printAt:
    push bp
    mov bp, sp
    push es
    push ax
    push cx
    push si
    push di

    mov ax, 0xb800
    mov es, ax

    ; compute offset = (row * 80 + col) * 2
    mov ax, [bp+8]      ; row
    mov bx, 80
    mul bx
    add ax, [bp+6]      ; col
    shl ax, 1
    mov di, ax

    mov si, [bp+12]     ; string address
    mov cx, [bp+10]     ; length
    mov ah, 0x0F        ; bright white

printLoop:
    mov al, [si]
    mov [es:di], ax
    add di, 2
    inc si
    loop printLoop
```

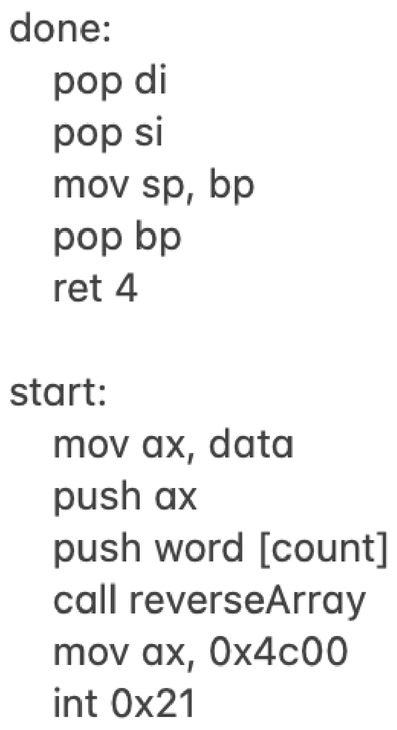


```
    pop di
    pop si
    pop cx
    pop ax
    pop es
    pop bp
    ret 8

start:
    mov ax, msg
    push ax
    push word [len]
    push word 10        ; row
    push word 15        ; col
    call printAt
    mov ax, 0x4c00
    int 0x21
```

Q6

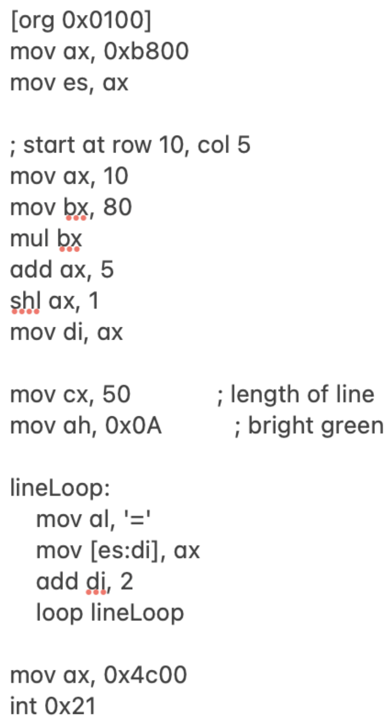


```
[org 0x0100]
jmp start

data: dw 1, 2, 3, 4, 5, 6
count: dw 6

reverseArray:
    push bp
    mov bp, sp
    sub sp, 2          ; local temp for swap
    push si
    push di

    mov si, [bp+6]     ; left pointer
    mov di, si
    mov ax, [bp+4]     ; count
    dec ax
    shl ax, 1
    add di, ax         ; right pointer

reverseLoop:
    cmp si, di
    jsge done

    mov ax, [si]
    mov [bp-2], ax
    mov ax, [di]
    mov [si], ax
    mov ax, [bp-2]
    mov [di], ax

    add si, 2
    sub di, 2
    jmp reverseLoop
```



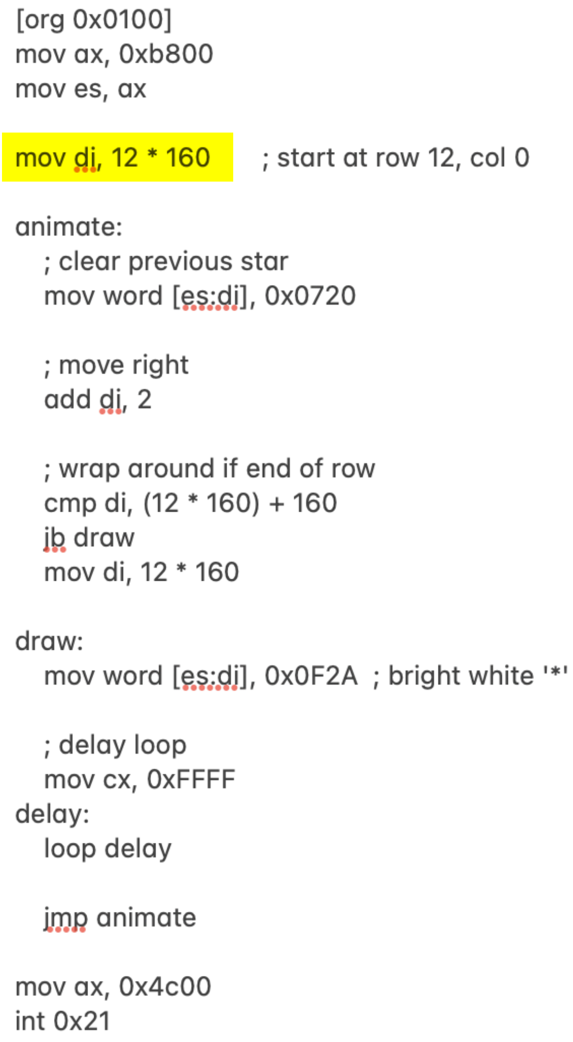
```
done:
   pop di
   pop si
   mov sp, bp
   pop bp
   ret 4

start:
   mov ax, data
   push ax
   push word [count]
   call reverseArray
   mov ax, 0x4c00
   int 0x21
```

Q7



```
[org 0x0100]
mov ax, 0xb800
mov es, ax

; start at row 10, col 5
mov ax, 10
mov bx, 80
mul bx
add ax, 5
shl ax, 1
mov di, ax

mov cx, 50            ; length of line
mov ah, 0x0A           ; bright green

lineLoop:
    mov al, '='
    mov [es:di], ax
    add di, 2
    loop lineLoop

mov ax, 0x4c00
int 0x21
```

Q8



```
[org 0x0100]
mov ax, 0xb800
mov es, ax

mov di, 12 * 160      ; start at row 12, col 0

animate:
   ; clear previous star
   mov word [es:di], 0x0720

   ; move right
   add di, 2

   ; wrap around if end of row
   cmp di, (12 * 160) + 160
   jb draw
   mov di, 12 * 160

draw:
   mov word [es:di], 0x0F2A  ; bright white '*'

   ; delay loop
   mov cx, 0xFFFF
delay:
   loop delay

   jmp animate

mov ax, 0x4c00
int 0x21
```