

Name: - ATRIJ ROY

ROLL NO: - 002311001086

SECTION: - IT A3 UG2

Jadavpur University  
Session 2024-25, Odd Semester  
Microprocessor Lab  
Paper Code: IT/S/222

## Assignment 2

---

**1. Write an Assembly Language Program to count the number of occurrence of 55H in a string of eight data bytes. The starting address of string is DS: 0030H. Store the count value in DS:0040H.**

```
.model small
.stack 100h
.data
.code
main proc
mov ax, @data
mov ds, ax
mov es, ax
mov al, 55h
mov cx, 0008h
mov di, 0030h
mov bl, 00h
l1: scasb
jnz l2
inc bl
```

```
l2:
loop l1
mov si, 0040h
mov [si], bl
int 03h
mov ah, 4ch
int 21h
main endp
end main
```

**2. Write an Assembly Language Program to find out the location where 55H is placed in a string of eight data bytes. The starting address of string is DS: 0030H.**

```
.model small
.stack 100h
.data
.code
main proc
mov ax, @data
mov ds, ax
mov es, ax
mov di, 0030h
mov al, 55h
mov cx, 0008h
mov si, 0040h
cld
l1:
scasb
jnz l2
dec di
mov [si], di
add si, 0002h
inc di
```

```
L2:
loop l1
int 03h
mov ah, 4ch
int 21h
main endp
end main
```

**3. Write an Assembly Language Program to compare two strings. The first string is stored from memory location DS: 0030H and the second sting is stored from DS: 0040H. Consider that the first byte of both strings contain the number of bytes contained in that string. If both strings are found equal, then show a value FFFFH in address DS: 0050H, otherwise show 1111H.**

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX
MOV ES, AX
MOV SI, 0030H
MOV DI, 0040H
MOV CL, [SI]
MOV CH, 00H
CLD
L2:
CMPSB
JNZ L1
LOOP L2
MOV AX, 0FFFFH
JMP L3
L1:
MOV AX, 01111H
```

```
L3:
MOV BX, 0050H
MOV [BX], AX
INT 03H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```

**4. Write an Assembly Language Program to check if a string of five data bytes is palindrome or not. The string is stored from memory location DS: 0030H. If the string is found to be palindrome then place FFFFH in addresses DS: 0040H otherwise place 1111H.**

```
.model small
.stack 100h
.data
.code
main proc
mov ax, @data
mov ds, ax
mov es, ax
mov ax, 0005h
mov si, 0030h
mov di, 0030h
add di, ax
dec di
mov bl, 02h
div bl
mov cl, al
mov ch, 00h
l1: mov al, [si]
mov bl, [di]
cmp al, bl
jnz l2
```

```

loop l1

mov ax, 0ffffh

jmp l3

l2: mov ax, 01111h

l3: mov bx, 0040h

mov [bx], ax

int 03h

mov ah, 4ch

int 21h

main endp

end main

```

- 5. Write an Assembly Language Program to count the number of positive and negative numbers present in a series of eight data bytes. The starting address of the series is DS: 0040H. Store the count value of positive number in DS: 0040H and count value of negative number in DS: 0041H.**

```

.model small

.stack 100h

data

.code

main proc

mov ax, @data

mov ds, ax

mov bx, 0000h

mov si, 0040h

mov cx, 0008h

l1:

mov al, [si]

rol al, 01h

inc si

jc l2

inc bh

jmp l3

```

```

l2:
inc bl

l3:
loop l1

mov si, 0040h
mov [si], bh
inc si
mov [si], bl
int 03h
mov ah, 4ch
21h
main endp
end main

```

- 6. Write an Assembly Language Program to separate the odd and even numbers from a series of 7 data bytes. The starting address of the series is DS: 0030H. Store the even numbers from DS: 0040H and the odd numbers from DS: 0050H.**

```

.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX
MOV ES, AX
MOV BX, 0030H
MOV SI, 0040H
MOV DI, 0050H
MOV CX, 0008H
L3:
MOV AL, [BX]
ROR AL, 01H
JC L1

```

```

ROL AL, 01H
MOV [SI], AL
INC SI
JMP L2
L1:
ROL AL, 01H
MOV [DI], AL
INC DI
L2:
INC BX
LOOP L3
INT 03H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN

```

7. **Write an Assembly Language Program to convert an 8-bit number stored in DS: 0030H into its equivalent ASCII value. Store the converted code from DS: 0050H.**

```

.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX
MOV SI, 0030H
MOV AL, [SI]
MOV AH, AL
AND AL, 0FH
CMP AL, 0AH

```

```

JC L1
ADD AL, 07
L1:
ADD AL, 30H
MOV BX, 0050H
MOV [BX], AL
MOV AL, AH
AND AL, 0F0H
MOV CL, 04H
ROL AL, CL
CMP AL, 0AH
JC L2
ADD AL, 07H
L2:
ADD AL, 30H
INC BX
MOV [BX], AL
INT 03H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN

```

**8. Write an Assembly Language Program to find out the square root of a number stored in DS: 0030H. Store the result in DS: 0040H.**

```

.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX

```



```

MOV SI, 0030H
MOV CL, 00H
MOV AL, [SI]
MOV BH, 00H
MOV AH, 01H
L3:
ADD BH, AH
CMP AL,AH
JZ L2
INC CL
ADD AH, 02H
JMP L3
L2:
MOV SI, 0040H
MOV [SI], CL
INT 03H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN

```

**9. Fibonacci series is defined as:**

**$F(i) = F(i-1) + F(i-2)$ ; for all  $i > 2$  with  $F(1) = F(2) = 1$**

**Write an Assembly language Program to generate the first ten elements of this sequence and store them from DS: 0030H.**

```

.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS, AX

```

```
MOV SI,0030H
MOV CX, 000AH
MOV [SI], 00H
INC SI
MOV [SI], 01H
L1:
MOV AL,[SI]
DEC SI
MOV AH,[SI]
INC SI
INC SI
ADD AL, AH
MOV [SI],AL
LOOP L1
INT 03H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```

---