**ASM Laboratory**



Name: **Soham Das**

Section: **A1**

Roll No: **002311001004**

**IT-UG2**

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.

**Code:**

.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

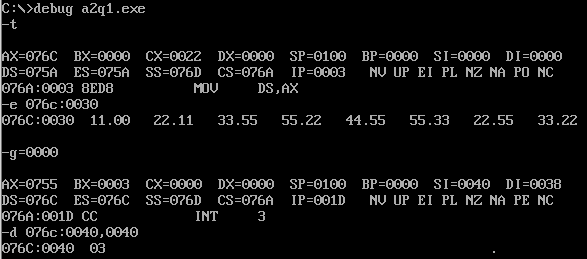
mov ah, 4ch

int 21h

main endp

end main

**Output:**



1. 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.

**Code:**

.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

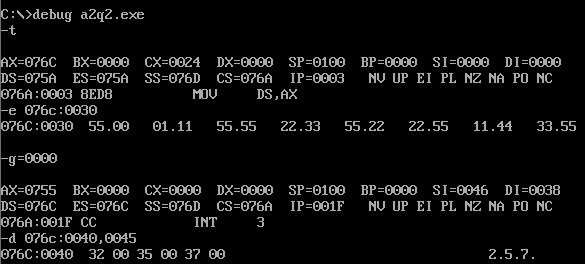
mov ah, 4ch

int 21h

main endp

end main

**Output:**



1. 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.

**Code**:

.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

l1:

cmpsb

jnz l2

loop l1

mov ax, 0ffffh

jmp l3

l2:

mov ax, 01111h

l3:

mov bx, 0050h

mov [bx], ax

int 03h

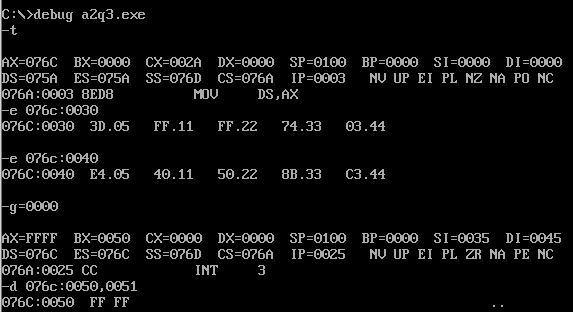
mov ah, 4ch

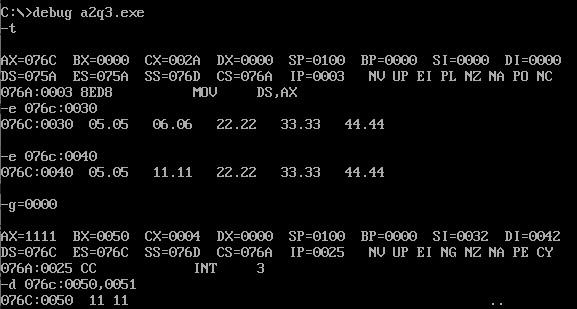
int 21h

main endp

end main

**Output:**





1. 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.

Code:

.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

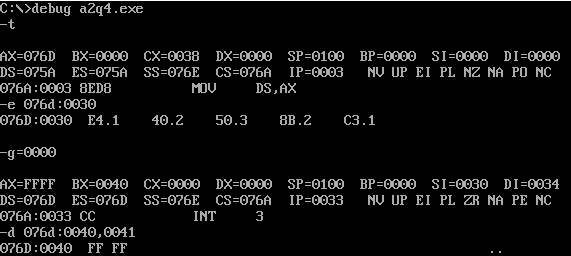
mov ah, 4ch

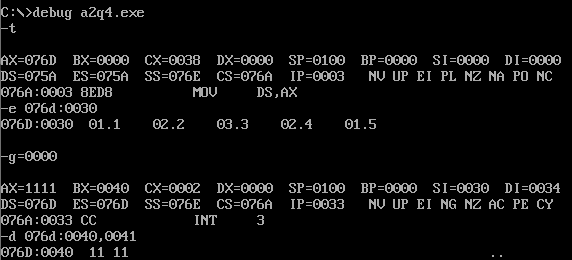
int 21h

main endp

end main

Output:





1. 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.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov bx, 0000h ; storing +ve in bh, -ve in bl

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

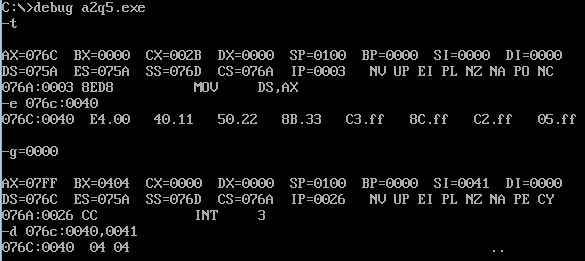
mov ah, 4ch

int 21h

main endp

end main

**Output:**



1. 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.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov es, ax

mov bx, 0030h

mov si, 0040h ;even

mov di, 0050h ;odd

mov cx, 0007h

l1: mov al, [bx]

ror al, 01h

inc bx

jnc l2

rol al, 01h

mov [di], al ;storing odd in si

inc di

jmp l3

l2: rol al, 01h

mov [si], al

inc si

l3: loop l1

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Output:**



1. 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.

**Code:**

.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, 09h

jc l2

add al, 07h

l2: add al, 30h

mov si, 0050h

mov [si], al

inc si

mov al, ah

and al, 0f0h

mov cl, 04h

rol al, cl

cmp al, 09h

jc l3

add al, 07h

l3: add al, 30h

mov [si], al

int 03h

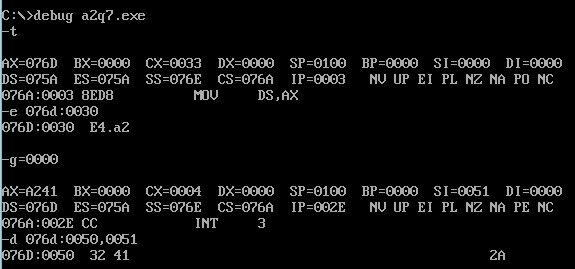
mov ah, 4ch

int 21h

main endp

end main

**Output:**



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

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov si, 0030h

mov al, [si]

mov bl, 01h

mov cl, 00h

l1: sub al, bl

das

add bl, 02h

daa

inc cl

cmp al, 00h

jz l2

jmp l1

l2: mov si, 0040h

mov [si], cl

int 03h

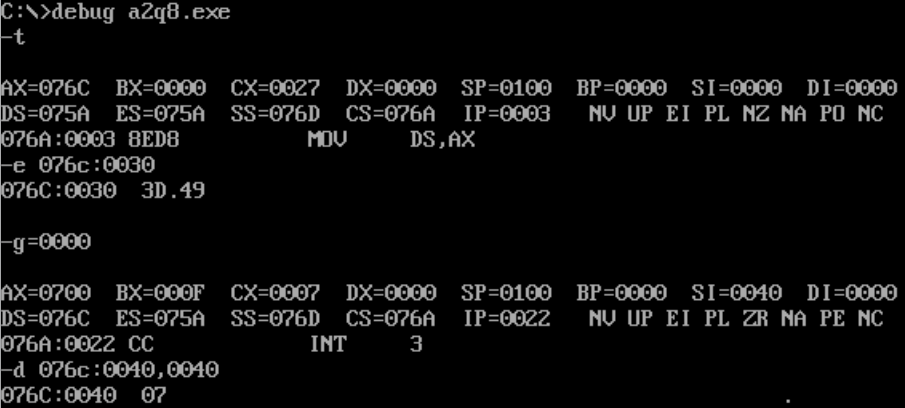
mov ah, 4ch

int 21h

main endp

end main

**Output:**



1. 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.

**Code:**

.model small

.stack 100h

.data

.code

main proc

mov ax, @data

mov ds, ax

mov cx, 000ah

mov al, 01h

mov bl, 01h

mov si, 0030h

l1:

mov [si], al

inc si

mov [si], bl

inc si

add al, bl

daa

xchg al, bl

add al, bl

daa

xchg al, bl

loop l1

int 03h

mov ah, 4ch

int 21h

main endp

end main

**Output:**

