1. To write and Execute ALP to add two 16-bit numbers.

Data Segment

a dw 4020h

b dw 2020h

sum dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bx,b

add ax,bx

mov sum,ax

mov ah,4ch

int 21h

Code ends

end Start

1. **To write and Execute ALP to add two 32-bit numbers.**

data Segment

op1 dd 12345678h

op2 dd 11111111h

ans dd ?

data ends

code Segment

assume cs:code,ds:data

start:

mov dx, data

mov ds, dx

mov ax, word ptr op1 ; lsb of number1 in ax

mov bx, word ptr op1+2 ; msb of number1 in bx

mov cx, word ptr op2 ; lsb of number2 in cx

mov dx, word ptr op2+2 ; msb of number1 in dx

add ax, cx ; add msb + msb + carry

mov word ptr ans, ax ; lsb answer

mov word ptr ans+2, bx ; msb answer

mov bx, word ptr ans+2 ; Result in reg bx

mov dh, 2

l1: mov ch, 04h ; Count of digits to be displayed

mov cl, 04h ; Count to roll by 4 bits

l2: rol bx, cl ; roll bl so that msb comes to lsb

mov dl, bl ; load dl with data to be displayed

and dl, 0fH ; get only lsb

cmp dl, 09 ; check if digit is 0-9 or letter A-F

jbe l4

add dl, 07 ; if letter add 37H else only add 30H

l4: add dl, 30H

mov ah, 02 ; INT 21H (Display character)

int 21H

dec ch ; Decrement Count

jnz l2

dec dh

cmp dh, 0

mov bx, word ptr ans ; display lsb of answer

jnz l1

mov ah, 4ch ; Terminate Program

int 21h

code ends

end Start

1. To write and Execute ALP to subtract two 16-bit numbers.

Data Segment

a dw 4020h

b dw 2020h

diff dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bx,b

sub ax,bx

mov diff,ax

mov ah,4ch

int 21h

Code ends

end Start

1. **To write and Execute ALP to subtract two 32-bit numbers.**

data segment

abc dd 9ABCDEF0h

def dd 12345678h

ghi dw ?

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

mov dl,00h

mov ax, word ptr abc

mov bx, word ptr def

sub ax,bx

mov word ptr ghi,ax

mov ax, word ptr abc+2

mov bx, word ptr def+2

sbb ax,bx

mov word ptr ghi+2,ax

jnc move

inc dl

move: mov byte ptr ghi+4,dl

int 3

code ends

end start

1. To write and Execute ALP to find sum of series of N Hex numbers.

Data Segment

array1 db 10h,20h,30h,40h,50h

sum dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

mov ax,0000h

lea si,array1

mov al,[si]

up: inc si

add al,[si]

jnc next

inc ah

next:dec cx

jnz up

mov sum,ax

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find sum of series of N BCD numbers.

Data Segment

array1 db 10h,20h,30h,40h,50h

sum dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

mov ax,0000h

lea si,array1

mov al,[si]

up: inc si

add al,[si]

daa

jnc next

inc ah

next:dec cx

jnz up

mov sum,ax

mov ah,4ch

int 21h

Code ends

end Start

1. **To write and Execute ALP to find sum of two blocks of N Hex numbers.**

Data Segment

array1 db 10h,20h,30h,40h,50h

sum dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

mov ax,0000h

lea si,array1

mov al,[si]

up: inc si

add al,[si]

jnc next

inc ah

next:dec cx

jnz up

mov sum,ax

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find sum of series of N BCD numbers.

Data Segment

array1 db 10h,20h,30h,40h,50h

sum dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

mov ax,0000h

lea si,array1

mov al,[si]

up: inc si

add al,[si]

daa

jnc next

inc ah

next:dec cx

jnz up

mov sum,ax

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to multiply two 08-bit Unsigned numbers.

Data Segment

a db 20h

b db 20h

prod dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov al,a

mov bl,b

mul bl

mov prod,ax

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to multiply two 16-bit Unsigned numbers.

Data Segment

a dw 2020h

b dw 1020h

prod dd ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bx,b

mul bx

mov word ptr [prod],ax

mov word ptr [prod+2],dx

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to multiply two 08-bit Signed numbers.

Data Segment

a db -20h

b db 20h

prod dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov al,a

mov bl,b

imul bl

mov prod,ax

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to multiply two 16-bit Signed numbers.

Data Segment

a dw 2020h

b dw -1020h

prod dd ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bx,b

imul bx

mov word ptr [prod],ax

mov word ptr [prod+2],dx

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Divide two Unsigned (16/8)

Data Segment

a dw 2030h

b db 50h

quo db ?

rem db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bl,b

div bl

mov quo,al

mov rem,ah

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Divide two Unsigned numbers(32/16)

Data Segment

a dd 10000000h

b dw 9999h

quo dw ?

rem dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,word ptr[a]

mov dx,word ptr[a+2]

mov bx,b

div bx

mov quo,ax

mov rem,dx

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Divide two Unsigned (16/8)

Data Segment

a dw 2030h

b db 50h

quo db ?

rem db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,a

mov bl,b

div bl

mov quo,al

mov rem,ah

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Divide two Unsigned numbers(32/16)

Data Segment

a dd 10000000h

b dw 9999h

quo dw ?

rem dw ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,word ptr[a]

mov dx,word ptr[a+2]

mov bx,b

div bx

mov quo,ax

mov rem,dx

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Add two BCD numbers.

Data Segment

a db 29h

b db 57h

res db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,0000h

mov al,a

mov bl,b

add al,bl

daa

jnc next

inc ah

next:

mov res,al

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Sub two BCD numbers.

Data Segment

a db 29h

b db 57h

res db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov ax,0000h

mov al,a

mov bl,b

sub al,bl

das

jnb next

dec ah

next:

mov res,al

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find smallest number from array of N numbers .

Data Segment

array1 db 10h,20h,30h,40h,50h

smallest db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

lea si,array1

mov al,[si]

up: inc si

cmp al,[si]

jb next

mov al,[si]

next:dec cx

jnz up

mov smallest,al

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find Largest number from array of N numbers .

Data Segment

array1 db 10h,20h,30h,40h,50h

largest db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

lea si,array1

mov al,[si]

up: inc si

cmp al,[si]

ja next

mov al,[si]

next:dec cx

jnz up

mov largest,al

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Arrange numbers in array in Ascending order.

Data Segment

array1 db 5,7,8,2

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

up1: lea si,array1

mov dx,cx

up2:mov al,[si]

cmp al,[si+1]

jna next

xchg al,[si+1]

mov [si],al

next:inc si

dec dx

jnz up2

loop up1

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to Arrange numbers in array in Descending order.

Data Segment

array1 db 5,7,8,2

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

mov cx,04h

up1: lea si,array1

mov dx,cx

up2:mov al,[si]

cmp al,[si+1]

jnb next

xchg al,[si+1]

mov [si],al

next:inc si

dec dx

jnz up2

loop up1

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find number of 0’s & 1’s from array of N numbers .

data segment

numbers db 1, 1, 1, 0, 0 ; Array of 5 numbers

num\_zeroes db 0 ; Number of zeroes in the array

num\_ones db 0 ; Number of ones in the array

data ends

; Code Segment

code segment

assume cs:code, ds:data

start:

mov dx, data ; Initialize Data Segment

mov ds, dx

mov ax, 0000h

mov cx, 5 ; Initialize loop counter to 5 (length of the array)

mov si, offset numbers ; Initialize pointer to start of array

count\_zeroes:

mov al, [si] ; Load number from array into AL

cmp al, 0 ; Compare number to 0

jne count\_ones ; If number is not 0, count as a 1

inc num\_zeroes

mov al, num\_zeroes ; If number is 0, increment number of zeroes

jmp next\_number ; Jump to next number in the array

count\_ones:

inc num\_ones ; If number is 1, increment number of ones

mov ah, num\_ones

next\_number:

inc si ; Move pointer to next number in the array

loop count\_zeroes ; Loop for remaining numbers in the array

mov ah, 4Ch ; Terminate the program

int 21h

code ends

end start

1. To write and Execute ALP to find EVEN & ODD numbers from array of N numbers .

Data Segment

array1 db 10,20,-30,-40,50

even1 db ?

odd1 db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

lea si,array1

mov bl,0h

mov bh,0h

mov cx,05h

up1:mov al,[si]

ror al,01

jc next1

inc bl

jmp next

next1:inc bh

next:inc si

dec dx

jnz up1

mov even1,bl

mov odd1,bh

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to find Positive & Negative numbers from array of N numbers

Data Segment

array1 db 10,20,-30,-40,50

pos db ?

nega db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov dx,Data

mov ds,dx

lea si,array1

mov bl,0h

mov bh,0h

mov cx,05h

up1:mov al,[si]

shr al,01

jc next1

inc bl

jmp next

next1:inc bh

next:inc si

dec dx

jnz up1

mov pos,bl

mov nega,bh

mov ah,4ch

int 21h

Code ends

end Start

1. To write and Execute ALP to transfer block of data from one memory block to another using string instruction.

disp macro msg

mov ah,09h

lea dx,msg

int 21h

endm

Data Segment

str1 db "Welcomes$"

str2 db 20 DUP('$')

msg1 db "Original String:$"

msg2 db "Copied String:$"

newl db 10,13,"$"

Data ends

Code Segment

assume CS:code,DS:data

Start:

mov dx,data

mov ds,dx

mov cx,0000h

lea si,str1

up: mov al,[si]

cmp al,'$'

je next

inc si

inc cx

jmp up

next:lea si,str1

lea di,str2

cld

up1:

movsb

loop up1

disp msg1

disp str1

disp newl

disp msg2

disp str2

mov ah,4ch

int 21h

code ends

end start

1. To write and Execute ALP to transfer block of data from one memory block to another without using string instruction.

disp macro msg

mov ah,09h

lea dx,msg

int 21h

endm

Data Segment

str1 db "Welcomes$"

str2 db 20 DUP('$')

msg1 db "Original String:$"

msg2 db "Copied String:$"

newl db 10,13,"$"

Data ends

Code Segment

assume CS:code,DS:data

Start:

mov dx,data

mov ds,dx

mov cx,0000h

lea si,str1

up: mov al,[si]

cmp al,'$'

je next

inc si

inc cx

jmp up

next:lea si,str1

lea di,str2

up1:

mov al,[si]

mov [di],al

inc si

inc di

loop up1

disp msg1

disp str1

disp newl

disp msg2

disp str2

mov ah,4ch

int 21h

code ends

end start

1. To write and Execute ALP to compare two strings using string instruction.
2. To write and Execute ALP to compare two strings without using string instruction.

disp macro msg

mov ah,09h

lea dx,msg

int 21h

endm

Data Segment

str1 db "Welcomes$"

str2 db "Welcomes$"

msg1 db "not same:$"

msg2 db "same:$"

newl db 10,13,"$"

Data ends

Code Segment

assume CS:code,DS:data

Start:

mov dx,data

mov ds,dx

mov cx,0000h

lea si,str1

up: mov al,[si]

cmp al,'$'

je next

inc si

inc cx

jmp up

next:lea si,str1

lea bx,str2

up1:

mov al,[si]

cmp al,[bx]

jne wrong

inc si

inc bx

loop up1

jmp correct

correct:

disp msg2

mov ah,4ch

int 21h

wrong:

disp msg1

mov ah,4ch

int 21h

code ends

end start

1. To write and execute ALP to find string Length & display string in reverse order.

disp macro msg

mov ah,09h

lea dx,msg

int 21h

endm

Data Segment

str1 db "bye$"

str2 db 20 DUP('$')

msg1 db "Original String:$"

msg2 db "Reversed String:$"

newl db 10,13,"$"

Data ends

Code Segment

assume CS:code,DS:data

Start:

mov dx,data

mov ds,dx

mov cx,0000h

lea si,str1

up: mov al,[si]

cmp al,'$'

je next

inc si

inc cx

jmp up

next:

lea di,str2

dec si

up1:

mov al,[si]

mov [di],al

dec si

inc di

loop up1

disp msg1

disp str1

disp newl

disp msg2

disp str2

mov ah,4ch

int 21h

code ends

end start