1. Sum and average

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter m1",0

m2 dbcr,lf,"enter m2",0

m3 dbcr,lf,"sum ",0

m4 dbcr,lf,"averg",0

a dw ?

b dw ?

avg dw 40 dup(?)

sum dw 10 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

output m2

inputs value,16

atoi value

mov b,ax

add ax,a

itoasum,ax

output m3

output sum

output m4

mov bl,2

div bl

itoaavg,al

output avg

mov ah,4ch

int 21h

code ENDS

END start

1. Area of rectangle

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter l",0

m2 dbcr,lf,"enter b",0

m3 dbcr,lf,"area ",0

a dw ?

b dw ?

area dw 40 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

output m2

inputs value,16

atoi value

mov b,ax

mov ax,a

mov bx,b

mul bx

itoaarea,ax

output m3

output area

mov ah,4ch

int 21h

code ENDS

END start

1. Greatest of two numbers

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter fir",0

m2 dbcr,lf,"enter sec",0

m3 dbcr,lf,"greatest ",0

a dw ?

b dw ?

num dw 40 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

output m2

inputs value,16

atoi value

mov b,ax

mov ax,a

mov bx,b

cmpax,bx

jg label1

output m3

itoanum,bx

output num

jmp quit

label1:output m3

itoanum,ax

output num

jmp quit

quit:mov ah,4ch

int 21h

code ENDS

END start

5>check if number is even or odd

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter number",0

m2 dbcr,lf,"odd number",0

m3 dbcr,lf,"even number ",0

a dw ?

b dw ?

num dw 40 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

mov ax,a

mov bx,2

cwd

div bx

cmp dx,0

je labeleven

output m2

jmp quit

labeleven:output m3

jmp quit

quit:mov ah,4ch

int 21h

code ENDS

END start

6.positive or negative

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter number",0

m2 dbcr,lf,"positive number",0

m3 dbcr,lf,"negative number ",0

a dw ?

b dw ?

num dw 40 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

mov ax,a

cmp ax,0

jge label1

output m3

jmp quit

label1:output m2

jmp quit

quit:mov ah,4ch

int 21h

code ENDS

END start

check if number divisible by 5

include io.h

data segment

crequ 0ah

lfequ 0dh

msg1 db 10,13,"enter number ",0

msg2 db 10,13,"number is not divisible by 5",0

msg3 db 10,13,"number is divisible by 5",0

a dw ?

value dw 10 dup(?)

result dw 10 dup(?)

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs value,16

atoi value

mov a,ax

mov bl,5

div bl

cmp ah,0h

jnz go

output msg3

jmp quit

go:output msg2

quit: mov al,00h

mov ah,4ch

int 21h

code ends

end start

check if number is even or odd

include io.h

data segment

crequ 0ah

lfequ 0dh

m1 dbcr,lf,"enter number",0

m2 dbcr,lf,"odd number",0

m3 dbcr,lf,"even number ",0

a dw ?

b dw ?

num dw 40 dup(?)

value dw 10 dup(?)

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

output m1

inputs value,16

atoi value

mov a,ax

mov ax,a

mov bx,2

cwd

div bx

cmp dx,0

je labeleven

output m2

jmp quit

labeleven:output m3

jmp quit

quit:mov ah,4ch

int 21h

code ENDS

END start

Smallest\_element\_array

include io.h

data segment

cr equ 0dh

lf equ 0ah

msg1 db cr,lf,&quot;enter number of elements in the array&quot;,0

msg2 db cr,lf,&quot;enter elements in the array&quot;,0

msg3 db cr,lf,&quot;the smallest element is&quot;,0

n dw ?; store length of array

e dw ? ;store indiviual element

num dw 40 dup(?)

array dw 40 dup(?)

buffer dw 40 dup(?); jus to take user input

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs num,40

atoi num

mov n,ax ;store len in variable n

mov dx,n ;transfer n to dx,for looping purpose

lea si,array ;source index points to first location of array

output msg2

label1: inputs buffer,40

atoi buffer

mov [si],ax ;moves to address pointed by si

add si,2 ;go to next address

dec dx ; bcz one element is already been taken

cmp dx,0 ;check if all elements are taken

jne label1; loop again to label1 if dx is not zero

lea si,array ;source index points to first location of array

mov dx,n ;load len of array in dx

mov bx,[si];keep bx to store the max number, initially first num is compared,bx will change

dec dx; bcz one element is taken

label2:

add si,2

mov cx,[si]

cmp cx,bx

jl label3

dec dx

cmp dx,0

jne label2

je label4

label3:

mov bx,cx

dec dx

cmp dx,0

jne label2

je label4

label4: output msg3

itoa buffer,bx

output buffer

mov ah,4ch

int 21h

code ends

end start

Largest\_element\_array

include io.h

data segment

cr equ 0dh

lf equ 0ah

msg1 db cr,lf,&quot;enter number of elements in the array&quot;,0

msg2 db cr,lf,&quot;enter elements in the array&quot;,0

msg3 db cr,lf,&quot;the largest element is&quot;,0

n dw ?; store length of array

e dw ? ;store indiviual element

num dw 40 dup(?)

array dw 40 dup(?)

buffer dw 40 dup(?); jus to take user input

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs num,40

atoi num

mov n,ax ;store len in variable n

mov dx,n ;transfer n to dx,for looping purpose

lea si,array ;source index points to first location of array

output msg2

label1: inputs buffer,40

atoi buffer

mov [si],ax ;moves to address pointed by si

add si,2 ;go to next address

dec dx ; bcz one element is already been taken

cmp dx,0 ;check if all elements are taken

jne label1; loop again to label1 if dx is not zero

lea si,array ;source index points to first location of array

mov dx,n ;load len of array in dx

mov bx,[si];keep bx to store the max number,initially first num is compared,bx will change

dec dx; bcz one element is taken

label2:

add si,2

mov cx,[si]

cmp cx,bx

jg label3

dec dx

cmp dx,0

jne label2

je label4

label3:

mov bx,cx

dec dx

cmp dx,0

jne label2

je label4

label4: output msg3

itoa buffer,bx

output buffer

mov ah,4ch

int 21h

code ends

end start

sum\_array\_elements

include io.h

data segment

cr equ 0dh

lf equ 0ah

msg1 db cr,lf,&quot;enter number of elements in the array&quot;,0

msg2 db cr,lf,&quot;enter elements in the array&quot;,0

msg3 db cr,lf,&quot;the sum is&quot;,0

msg4 db cr,lf,&quot;the array is&quot;,0

num dw 40 dup(?)

n dw ?

array dw 40 dup(?)

sum dw 40 dup(?)

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs num,40

atoi num

mov n,ax

mov dx,n

lea si,array

output msg2

label1: inputs num,40

atoi num

mov [si],ax

add si,2

dec dx

cmp dx,0

jne label1

mov dx,n

output msg4

lea si,array

label2:

mov ax,[si]

itoa sum,ax

output sum

add si,2

dec dx

cmp dx,0

jne label2

lea si,array

mov bx,0

mov dx,n

label3:

mov ax,[si]

add bx,ax

add si,2

dec dx

cmp dx,0

jne label3

output msg3

itoa sum,bx

output sum

mov ah,4ch

int 21h

code ends

end start

Fibonacci series

include io.h

data segment

cr equ 0dh

lf equ 0ah

msg1 db cr,lf,&quot;enter the number&quot;,0

n dw ?

result dw 6 dup(?)

value dw 40 dup(?)

a dw 6 dup(?)

b dw 6 dup(?)

c dw 6 dup(?)

d dw 6 dup(?)

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs value,16

atoi value

mov n,ax

mov cx,2

mov a,0

mov b,1

mov ax,a

itoa d,ax

output d

mov ax,b

itoa d,ax

output d

mov ax,b

add ax,a

mov c,ax

label1:itoa d,ax

output d

inc cx

mov bx,b

mov a,bx

mov bx,c

mov b,bx

mov ax,b

add ax,a

mov c,ax

cmp cx,n

jnz label1

mov ah,4ch

int 21h

code ends

end start

factorial

include io.h

data segment

cr equ 0dh

lf equ 0ah

msg1 db cr,lf,&quot;enter the number&quot;,0

n dw ?

result dw 6 dup(?)

value dw 40 dup(?)

f dw ?

data ends

code segment

assume cs:code ,ds:data

start:

mov ax,data

mov ds,ax

output msg1

inputs value,16

atoi value

mov f,ax

mov cx,f

mov ax,1

label1:imul cx

dec cx

jnz label1

itoa result,ax

output result

mov ah,4ch

int 21h code ends end start