**UVA 10071**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0 ;all var init

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov ax,bx ;logic

mov bx,2

mul bp

mul bx

call output ;output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

ENDP MAIN

**UVA 10812**

org 100h

.model small

.stack 100h

.data

w db 'impossible $'

x dw 0

y dw 10

z dw 0

t dw 2

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov temp,0

cmp te,0 ;comparing test case

je exit

call input

mov bx,temp

mov temp,0

call input

cmp bx,temp

jl l1

mov ax,temp

add ax,bx

mov dx,0

div t

cmp dx,1

je l1

call output

mov ax,bx

sub ax,temp

mov dx,0

div t

mov z,0

call output

jmp l2

l1:

lea dx,w

mov ah,9

int 21h

call newline

l2:

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10970**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov ax,bx ;logic

mul bp

dec ax

call output ;output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11150**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov bp,3

mov cx,bx

mov ax,bx

l: ;logic

cmp cx,bp

jl ck

mov dx,0

xchg ax,cx

div bp

add cx,ax

add ax,dx

xchg ax,cx

jmp l

ck:

cmp cx,2

jne output

inc ax

call output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11854**

org 100h

.model small

.stack 100h

.data

w db 'wrong $'

r db 'right $'

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov temp,0

mov z,0

call input ;1st side

mov bx,temp

mov temp,0

call input

mov bp,temp ;2nd side

mov temp,0

call input

mov cx,temp ;3rd side

cmp bx,0

je exit

mov ax,bx ;a\*a

mul bx

mov bx,ax

mov ax,bp ;b\*b

mul bp

mov bp,ax

mov ax,cx ;c\*c

mul cx

mov cx,ax

cmp bx,bp ;making 1st side greater

jg l1

xchg bx,bp

l1:

cmp bx,cx

jg l2

xchg bx,cx

l2:

add bp,cx

cmp bx,bp ;compare a\*a=b\*b+c\*c

je right

lea dx,w ;output 'wrong'

mov ah,9

int 21h

call newline

jmp start

right:

lea dx,r ;output 'right'

mov ah,9

int 21h

call newline

jmp start

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10696**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

cmp bx,0 ;logic

je exit

mov ax,bx

cmp ax,100

jg l

mov ax,91

call output ;output

l:

sub ax,y

call output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 12577**

org 100h

.model small

.stack 100h

.data

hajj db 'Hajj-e-Akbar',10,13,'$'

umrah db 'Hajj-e-Asghar',10,13,'$'

.code

main proc

start:

mov ah,1

int 21h

mov bl,al

ex:

mov ah,1

int 21h

cmp al,13

jne ex

cmp bl,'\*'

je exit

call newline

cmp bl,'H'

je haj

lea dx,umrah

mov ah,9

int 21h

jmp start

haj:

lea dx,hajj

mov ah,9

int 21h

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10055**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

cmp bx,bp

jl Less

sub bx,bp ;sub 2nd from 1st

mov ax,bx

call output

Less:

sub bp,bx ;sub 1st from 2nd

mov ax,bp

call output ;output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10783**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

t dw 2

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

cmp te,0 ;comparing test case

je exit

mov bp,0

mov temp,0

mov z,0

call input ;input 1st

mov bx,temp

mov temp,0

call input ;input 2nd

L:

cmp bx,temp ;adding odd sum

jg l2

mov ax,bx

mov dx,0

div t

cmp dx,0

jne adding

inc bx

adding:

add bp,bx

add bx,2

jmp L

l2:

mov ax,bp

call output

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10346**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov cx,bx

mov ax,bx

l: ;logic

cmp cx,bp

jl output

mov dx,0

xchg ax,cx

div bp

add cx,ax

add ax,dx

xchg ax,cx

jmp l

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10079**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov x,0

mov temp,0

mov z,0

call input ;1st input

mov ax,temp

cmp ax,0

jl exit

mov bx,temp

mul bx

add ax,bx

add ax,2

mov bx,2

div bx

call output ;output

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 136**

org 100h

.stack 100h

.model small

.data

x db 'The 1500',39,'th ugly number is 859963392.',10,13,'$'

.code

main proc

mov ax,@data

mov ds,ax

lea dx,x

mov ah,9

int 21h

exit:

endp main

**UVA 113**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

mov ax,bx

mov cx,1

logic: ;logic

inc cx

mul bx

cmp ax,bp

jne logic

mov ax,cx

call output

j:

call newline

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call space

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

space proc

mov ah,2

mov dl,32

int 21h

ret

exit:

mov ah,4ch

int 21h

end main

**UVA 424**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

mov bp,0

start:

mov temp,0

mov z,0

call input ;taking input

mov ax,temp

add bp,ax ;sum till found '0'

cmp ax,0

jne start

mov ax,bp

call output

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp exit

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 483**

org 100h

.model small

.stack 100h

.data

z dw 0

t dw 0

ar db 100 dup(?)

.code

main proc

start:

mov di,0

mov z,0

mov t,0

st:

mov ah,1

int 21h

cmp al,13 ;comp with enter

jne l

inc t

mov al,32

l:

cmp al,32 ;comp with space

je psh

mov ah,0

inc z

push ax

jmp st

psh:

mov cx,z

mov z,0

lp:

pop dx ;store rev in ary

mov ar[di],dl

inc di

loop lp

mov ar[di],32

inc di

cmp t,0

je st

call newline

mov cx,di

dec cx

mov di,0

ot: ;output array

mov ah,2

mov dl,ar[di]

int 21h

inc di

loop ot

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 1124**

org 100h

.model small

.stack 100h

.data

ar db 100 dup(?)

.code

main proc

start:

mov di,0

call input

mov cx,di

mov di,0

call output

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ar[di],al

inc di

jmp input

output proc

print:

mov dl,ar[di]

int 21h

inc di

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10110**

org 100h

.model small

.stack 100h

.data

ye db 'yes$'

n db 'no$'

x dw 0

y dw 10

temp dw 0

.code

main proc

mov ax,@data ;load data

mov ds,ax

start:

call input

mov bx,temp

mov temp,0

cmp bx,0

je exit

mov cx,1

lp:

mov ax,cx

mul cx

cmp ax,bx

je yes

cmp ax,bx

jg no

inc cx

jmp lp

yes: ;yes output

lea dx,ye

mov ah,9

int 21h

call newline

jmp start

no: ;no output

lea dx,n

mov ah,9

int 21h

call newline

jmp start

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 10591**

org 100h

.model small

.stack 100h

.data

h db 'Happy Number$'

uh db 'Unhappy Number$'

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

.code

main proc

mov ax,@data ;load data

mov ds,ax

call input ;input test case

mov bx,temp

mov t,bx

start:

cmp t,0

je exit

mov temp,0

mov z,0

call input ;input number

mov bx,temp

cmp bx,7

je happy

mov ax,bx

l: ;squaring until sum<10

mov bp,0

lp:

mov dx,0

div y

mov bx,ax

mov ax,dx

mul dx

add bp,ax

mov ax,bx

cmp bx,0

jne lp

mov ax,bp

cmp ax,10

jge l

cmp ax,1

jne unhappy

happy: ;happy output

lea dx,h

mov ah,9

int 21h

call newline

jmp j

unhappy: ;unhappy output

lea dx,uh

mov ah,9

int 21h

call newline

j:

dec t

jmp start

input proc

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11044**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

t dw 3

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov z,0

mov temp,0

cmp te,0 ;comp test case

je exit

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

mov ax,bx ;logic

mov dx,0

div t

mov bx,ax

mov ax,bp

mov dx,0

div t

mul bx

call output

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11172**

org 100h

.model small

.stack 100h

.data

b db '> $'

s db '< $'

e db '= $'

x dw 0

y dw 10

z dw 0

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov temp,0

cmp te,0 ;comp test case

je exit

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

cmp bx,bp ;comparing

je eq

cmp bx,bp

jg bg

lea dx,s ;output '<'

mov ah,9

int 21h

jmp L

eq: ;output '='

lea dx,e

mov ah,9

int 21h

jmp L

bg: ;output '>'

lea dx,b

mov ah,9

int 21h

L:

call newline

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11192**

org 100h

.model small

.stack 100h

.data

z db 0

t db 0

ar db 100 dup(?)

.code

main proc

mov ax,@data ;load data

mov ds,ax

start:

mov t,0

mov z,0

mov ah,1 ;input n to reverse

int 21h

mov bl,al

sub bl,48

cmp bl,0

mov bh,0

je exit

call newline

mov di,0

st: ;logic

mov cx,bx

inc z

mov ah,1

int 21h

cmp al,13

je j

mov ah,0

push ax

cmp bl,z

je arrayin

jmp st

arrayin: ;store in ary in rev

mov z,0

mov dx,0

pop dx

mov ar[di],dl

inc di

loop arrayin

jmp st

j:

call newline

mov cx,di

mov di,0

arrayout: ;show output

mov ah,2

mov dl,ar[di]

int 21h

inc di

loop arrayout

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11313**

org 100h

.model small

.stack 100h

.data

g db 'cannot do this $'

x dw 0

y dw 10

z dw 0

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov temp,0

mov z,0

cmp te,0 ;comp test case

je exit

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

dec bx

dec bp

mov dx,0

mov ax,bx

div bp

cmp dx,0

jne l

call output

jmp j

l:

lea dx,g

mov ah,9

int 21h

call newline

j:

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11332**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov temp,0

mov z,0

mov bp,0

call input

mov ax,temp

cmp ax,0

je exit

l:

cmp ax,0

je f

mov dx,0

div y

add bp,dx

jmp l

f:

mov ax,bp

mov bp,0

cmp ax,y

jnl l

call output

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11364**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

.code

main proc

mov ax,@data

mov ds,ax

call input ;input test case

mov bx,temp

mov t,bx

start:

mov z,0

mov temp,0

cmp t,0 ;comp test case

je exit

call input

mov cx,temp ;input n

mov temp,0

dec cx

call input

mov bx,temp ;initialy min

mov bp,temp ;initialy max

ad:

mov temp,0

call input

cmp bx,temp ;final min

jl l

mov bx,temp

l:

cmp bp,temp ;final max

jg g

mov bp,temp

g:

loop ad

sub bp,bx ;2\*(max-min)

mov ax,bp

mov bx,2

mul bx

call output

dec t

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11388**

org 100h

.model small

.stack 100h

.data

g db '-1 $'

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

.code

main proc

mov ax,@data

mov ds,ax

call input ;input test case

mov bx,temp

mov t,bx

start:

mov temp,0

cmp t,0 ;comp test case

je exit

call input

mov bx,temp ;1st input

mov temp,0

call input

mov bp,temp ;2nd input

mov temp,0

mov dx,0

mov ax,bp ;logic

div bx

cmp dx,0

jne l

mov ax,bx

mov z,0

call output

mov ax,bp

mov z,0

call output

jmp j

l:

lea dx,g

mov ah,9

int 21h

call newline

j:

dec t

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11462**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

n dw 0

ar dw 100 dup(?)

.code

main proc

start:

mov temp,0

call input ;input n

mov ax,temp

cmp ax,0

je exit

dec ax

add ax,ax

mov n,ax

mov di,0

arrayin: ;array input

mov temp,0

call input

mov ax,temp

mov ar[di],ax

cmp di,n

je s

add di,2

jmp arrayin

s: ;sort logic

mov di,0

mov bp,0

sort: ;1st loop

add bp,di

add bp,2

s1: ;2nd loop

mov bx,ar[bp]

cmp ar[di],bx

jle s2

mov cx,ar[di] ;swap function

mov ar[di],bx

mov ar[bp],cx

s2:

add bp,2

cmp bp,n

jle s1

mov bp,0

add di,2

cmp di,n

jl sort

mov di,0

arrayout: ;array output

mov z,0

mov ax,ar[di]

call output

cmp di,n

je st

add di,2

jmp arrayout

st:

call newline

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call space

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

space proc

mov ah,2

mov dl,32

int 21h

ret

exit:

endp main

**UVA 11479**

org 100h

.model small

.stack 100h

.data

a db 'Invalid $'

b db 'Equilateral $'

c db 'Isosceles $'

d db 'Scalene $'

x dw 0

y dw 10

temp dw 0

t dw 0

.code

main proc

mov ax,@data

mov ds,ax

call input ;input test case

mov ax,temp

mov t,ax

start:

mov temp,0

cmp t,0 ;com test case

je exit

call input

mov bx,temp

mov temp,0

call input

mov bp,temp

mov temp,0

call input

mov cx,temp

mov temp,0

cmp bx,bp

jle g1

xchg bx,bp

g1:

cmp bp,cx

jle g2

xchg bp,cx

g2:

mov ax,bx

add ax,bp

cmp ax,cx

jle inv

cmp bx,bp

je l1

cmp bp,cx

je iso

jmp sc

l1:

cmp bp,cx

jne iso

lea dx,b

mov ah,9

int 21h

jmp j

iso:

lea dx,c

mov ah,9

int 21h

jmp j

sc:

lea dx,d

mov ah,9

int 21h

jmp j

inv:

lea dx,a

mov ah,9

int 21h

j:

dec t

call newline

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11727**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

.code

main proc

mov ax,@data

mov ds,ax

call input ;input test case

mov bx,temp

mov t,bx

start:

mov z,0

mov temp,0

cmp t,0 ;comparing test case

je exit

call input ;1st input

mov bp,temp

mov temp,0

call input

mov bx,temp ;2nd input

mov temp,0

call input

mov cx,temp ;3rd input

mov temp,0

cmp bp,bx ;logic

jg g ;a<b<c (b)

xchg bp,bx

g:

cmp bp,cx

jg l

xchg bp,cx

l:

cmp bx,cx

jg j

xchg bx,cx

j:

mov ax,bx

call output

dec t

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11799**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

n dw 0

.code

main proc

mov ax,@data

mov ds,ax

call input ;input test case

mov bx,temp

mov t,bx

start:

cmp t,0 ;comp test case

je exit

mov temp,0

call input ;input n

mov ax,temp

mov n,ax

mov bp,0

lp:

cmp n,0

je j

mov temp,0 ;logic of big

call input

cmp bp,temp

jge d1

xchg bp,temp

d1:

dec n

jmp lp

j:

mov ax,bp

mov z,0

call output

dec t

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11805**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov z,0

mov temp,0

cmp te,0 ;comp test case

je exit

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

call input ;3rd input

mov cx,temp

mov temp,0

mov ax,cx ;logic

add ax,bp

mov dx,0

div bx

cmp dx,0

je l

mov ax,dx

call output

dec te

jmp start ;jump starting

l:

mov ax,bx

call output

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11877**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov temp,0

mov z,0

call input ;taking input

mov ax,temp

cmp ax,0

je exit

mov bx,2 ;logic

mov dx,0

div bx

call output

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 11984**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

t dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov t,bx

start:

mov temp,0

mov z,0

cmp t,0 ;comp test case

je exit

call input

mov bx,temp ;1st input

mov temp,0

call input

mov bp,temp ;2nd input

mov temp,0

mov ax,bx ;logic

mov cx,5

mov temp,9

mov dx,0

mul temp

div cx

add ax,bp

mul cx

mov dx,0

div temp

call output

dec t

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

ret

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 12149**

org 100h

.model small

.stack 100h

.data

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov temp,0

mov z,0

call input ;taking input

mov ax,temp

cmp ax,0

je exit

mov bp,0

mov bx,1

L:

cmp bx,temp

jg ot

mov ax,bx

mul bx

add bp,ax

inc bx

jmp L

ot:

mov ax,bp

call output

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 12372**

org 100h

.model small

.stack 100h

.data

g db 'good $'

b db 'bad $'

c dw 20

x dw 0

y dw 10

z dw 0

temp dw 0

te dw 0

.code

main proc

call input ;input test case

mov bx,temp

mov te,bx

start:

mov temp,0

cmp te,0 ;comp test case

je exit

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

call input ;3rd input

mov cx,temp

mov temp,0

cmp bx,c ;comparing logic

jg l

cmp bp,c

jg l

cmp cx,c

jg l

lea dx,g ;output 'good'

mov ah,9

int 21h

jmp ed

l:

lea dx,b ;output 'bad'

mov ah,9

int 21h

ed:

call newline

dec te

jmp start ;jump starting

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 12478**

org 100h

.model small

.stack 100h

.data

g db 'kABIR',10,13,'$'

.code

main proc

lea dx,g

mov ah,9

int 21h

exit:

main endp

**UVA 12646**

org 100h

.model small

.stack 100h

.data

aa db 'A$'

bb db 'B$'

cc db 'C$'

st db '\*$'

.code

main proc

start:

mov ah,1

int 21h

mov bl,al ;1st input

int 21h

mov cl,al ;2nd input

int 21h

mov ch,al ;3rd input

call newline

cmp bl,cl ;logic

je l1

cmp cl,ch

je l2

cmp bl,ch

je l3

l1:

cmp bl,ch

jne c

jmp s

l2:

cmp bl,cl

jne a

jmp s

l3:

cmp bl,cl

jne b

jmp s

a:

lea dx,aa ;output 'A'

mov ah,9

int 21h

jmp j

b:

lea dx,bb ;output 'B'

mov ah,9

int 21h

jmp j

c:

lea dx,cc ;output 'C'

mov ah,9

int 21h

jmp j

s:

lea dx,st ;output '\*'

mov ah,9

int 21h

j:

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main

**UVA 12468**

org 100h

.model small

.stack 100h

.data

c dw 50

x dw 0

y dw 10

z dw 0

temp dw 0

.code

main proc

start:

mov temp,0

mov z,0

call input ;1st input

mov bx,temp

mov temp,0

call input ;2nd input

mov bp,temp

mov temp,0

cmp bx,bp ;swap if (a<b)

jg xc

xchg bx,bp

xc:

mov ax,bx

sub ax,bp

cmp ax,c ;compare (a-b)>50

jg l

call output

l:

mov ax,100 ;if true

add ax,bp

sub ax,bx

call output

input proc ;input

mov ah,1

int 21h

cmp al,13

jne sum

call newline

ret

sum:

mov ah,0

sub al,48

mov x,ax

mov ax,temp

mul y

add ax,x

mov temp,ax

jmp input

output proc ;output

mov dx,0

div y

push dx

inc z

mov cx,0

mov cx,z

cmp ax,0

je print

jmp output

print:

mov ah,2

pop dx

add dl,48

int 21h

loop print

call newline

jmp start

newline proc ;newline

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

ret

exit:

endp main