**problem : 458**

**--------------------**

.model small

.stack 100h

.data

a db 100 dup(1)

.code

main proc

mov ax,@data

mov ds,ax

loop1:

mov si,0

loop2:

mov ah,1

int 21h

mov a[si],al

inc si

cmp al,13

jne loop2

call newline

mov si,0

loop3:

sub a[si],7

mov ah,2

mov dl,a[si]

int 21h

inc si

cmp a[si],13

jne loop3

call newline

jmp loop1

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 1124**

**----------------------**

.model small

.stack 100h

.data

a db 100 dup(-1)

.code

main proc

mov ax,@data

mov ds,ax

loop1:

mov si,0

mov ah,1

loop2:

int 21h

mov a[si],al

inc si

cmp al,13

jne loop2

call newline

mov si,0

mov ah,2

loop3:

mov dl,a[si]

int 21h

inc si ;si++

cmp a[si],13

jne loop3

call newline

jmp loop1

exit:

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 10055**

**------------------------**

.model small

.stack 100h

.code

main proc

loop1:

mov ah,1

int 21h

mov bl,al

mov ah,2

mov dl,' '

int 21h

mov ah,1

int 21h

mov bh,al

cmp bh,bl

jg label1

sub bl,bh

mov bh,bl

add bh,48

jmp label2

label1:

sub bh,bl

add bh,48

label2:

;for output

mov ah,2

mov dl,13

int 21h

mov dl,10

int 21h

mov dl,bh

int 21h

mov dl,13

int 21h

mov dl,10

int 21h

jmp loop1

mov ah,4ch

int 21h

main endp

end main

ret

**Problem : 11313**

**------------------------**

.model small

.stack 100h

.data

a dw ?

b dw ?

s db "cannot do this $"

.code

main proc

mov ax,@data

mov ds,ax

call taking\_input

dec ax

mov a,ax

call newline

call taking\_input

dec ax

mov b,ax

call newline

mov ax,a

mov bx,b

mov dx,0

div bx

cmp dx,0

jne soutput

mov ax,a

mov bx,b

div bx

call printing\_output

jmp exit

soutput:

mov ah,9

lea dx,s

int 21h

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 10071**

**----------------------**

.model small

.stack 100h

.data

.code

main proc

loop1:

mov ah,1

int 21h

mov bl,al

sub bl,48

mov ah,2

mov dl, ' '

int 21h

mov ah,1

int 21h

mov bh,al

sub al,48

mov cl,1

cmp bl,1

je level

loop2:

add bh,al

inc cl

cmp cl,bl

jl loop2

level:

mov al,bh

sub al,48

add bh,al

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

mov dl,bh

int 21h

mov dl,0ah

int 21h

mov dl,0dh

int 21h

jmp loop1

mov ah,4ch

int 21h

main endp

end main

**problem : 10079**

**----------------------**

.model small

.stack 100h

.code

main proc

start:

call scan

cmp dl, 0

je exit

mov cx, 0

mov cl, dl

mov ax, 1

sum:

add al, cl

loop sum

call print

mov dl, 10

int 21h

mov dl, 13

int 21h

jmp start

exit:

ret

main endp

scan proc

mov dx, 0

input:

mov ah, 1

int 21h

cmp al, ' '

je end

cmp al, 13

je end

push ax

mov al, 10

mul dl

mov dl, al

pop ax

sub al, '0'

add dl, al

jmp input

end:

ret

scan endp

print proc

xor cx, cx

loop1:

cwd

mov bx, 10

idiv bx

push dx

inc cx

cmp ax, 0

jg loop1

loop2:

pop dx

add dx, '0'

mov ah, 2

int 21h

loop loop2

ret

print endp

**problem : 10110**

**------------------------**

.model small

.stack 100h

.data

yes db 'yes$'

no db 'no$'

n db ?

.code

main proc

mov ax,@data

mov ds,ax

loop1:

mov n,1

mov ah,1

int 21h

and bx,0

mov bl,al

sub bl,48

cmp bl,0

je exit

call newline

loop2:

and ax,0

mov al,n

mul n

cmp al,bl

je print\_yes

jg print\_no

inc n

jmp loop2

print\_yes:

mov ah,9

lea dx,yes

int 21h

call newline

jmp loop1

print\_no:

mov ah,9

lea dx,no

int 21h

call newline

jmp loop1

exit:

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,0dh

int 21h

mov dl,0ah

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 12461**

.model small

.stack 100h

.data

n dw ?

s db "1/2 $"

.code

main proc

mov ax,@data

mov ds,ax

call taking\_input

mov n,ax

call newline

mov ah,9

lea dx,s

int 21h

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 12149**

.model small

.stack 100h

.data

n dw ?

c dw ?

.code

main proc

mov ax,@data

mov ds,ax

mov c,0

call taking\_input

mov n,ax

call newline

mov cx,n

inc cx

mov n,cx

level:

mov cx,n

dec cx

mov n,cx

cmp cx,0

je output

mov bx,n

mov ax,n

mul bx

mov bx,c

add bx,ax

mov c,bx

jmp level

output:

mov ax,c

call printing\_output

call newline

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 10696**

**-----------------------**

.model small

.stack 100h

.code

main proc

start:

call scan

cmp dl, 0

je exit

cmp dl, 100

jg greater

mov dl, 91

jmp result

greater:

sub dl, 10

result:

mov ax, 0

mov al, dl

call print

mov dl, 10

int 21h

mov dl, 13

int 21h

jmp start

exit:

ret

main endp

scan proc

mov dx, 0

input:

mov ah, 1

int 21h

cmp al, ' '

je end

cmp al, 13

je end

push ax

mov al, 10

mul dl

mov dl, al

pop ax

sub al, '0'

add dl, al

jmp input

end:

ret

scan endp

print proc

xor cx, cx

loop1:

cwd

mov bx, 10

idiv bx

push dx

inc cx

cmp ax, 0

jg loop1

loop2:

pop dx

add dx, '0'

mov ah, 2

int 21h

loop loop2

ret

print endp

**problem : 13059**

.model small

.stack 100h

.data

t dw ?

.code

main proc

mov ax,@data

mov ds,ax

call taking\_input

mov t,ax

call newline

mov ax,t

dec ax

call printing\_output

call newline

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 13046**

.model small

.stack 100h

.data

r dw ?

t dw ?

z dw ?

a dw ?

n dw ?

k dw ?

.code

main proc

mov ax,@data

mov ds,ax

call taking\_input

mov n,ax

mov a,ax

mov t,ax

call newline

call taking\_input

mov k,ax

call newline

level:

mov cx,a

cmp cx,k

jl output

mov ax,a

mov bx,k

mov dx,0

div bx

mov r,ax

add ax,t

mov t,ax

add dx,r

mov a,dx

jmp level

output:

mov ax,t

call printing\_output

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 13025**

.model small

.stack 100h

.data

str db "may 29, 2013 wednesday $"

.code

main proc

mov ax,@data

mov ds,ax

lea dx,str

mov ah,9

int 21h

mov ah,4ch

int 21h

main endp

end main

**problem : 13012**

.model small

.stack 100h

.data

t dw ?

n dw ?

x dw ?

sum dw ?

.code

main proc

mov ax,@data

mov ds,ax

mov sum,0

call taking\_input

mov n,ax

call newline

mov t,5

mov cx,t

inc cx

mov t,cx

level:

mov cx,t

dec cx

mov t,cx

cmp cx,0

je output

call taking\_input

mov x,ax

call newline

mov ax,x

cmp ax,n

je cal

jmp level

cal:

mov bx,sum

inc sum

jmp level

output:

mov ax,sum

call printing\_output

call newline

exit:

mov ah,4ch

int 21h

main endp

newline proc

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

ret

newline endp

taking\_input proc

mov cx,0

input:

mov ah,1

int 21h

cmp al,13

jne calc

mov ax,cx

ret

calc:

sub al,48

mov ah,0

mov bx,ax

mov dx,10

mov ax,cx

mul dx

add ax,bx

mov cx,ax

jmp input

taking\_input endp

printing\_output proc

mov cx,0

print:

mov dx,0

mov bx,10

div bx

push dx

inc cx

cmp ax,0

jne print

printf:

mov ah,2

pop dx

add dl,48

int 21h

loop printf

ret

printing\_output endp

end main

**problem : 10783**

**----------------------**

.model small

.stack 100h

.data

.code

main proc

mov ah,1

int 21h

mov cl,al

sub cl,30h

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

loop:

mov ah,1

int 21h

mov bh,al

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

mov ah,1

int 21h

mov bl,al

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

sub bh,30h

sub bl,30h

mov al,bh

add al,48

loop2:

sub al,2

cmp al,1

jg loop2

je odd

jl even

odd:

mov ch,0

jmp loop3

even:

add bh,1

mov ch,0;

loop3:

add ch,bh

add bh,2

cmp bh,bl

jle loop3

add ch,48

mov dl,ch

int 21h

mov dl,10

int 21h

mov dl,13

int 21h

dec cl

cmp cl,0

jg loop

exit:

mov ah,4ch

int 21h

main endp

end main

**problem : 10812**

**-----------------------**

.model small

.stack 100h

.data

imp db 'impossible', '$'

nl db 10, 13, '$'

a db ?

b db ?

.code

main proc

mov ax, @data

mov ds, ax

call scan

mov cx, 0

mov cl, dl

lea dx, nl

mov ah, 9

int 21h

testcase:

push cx

call scan

mov [a], dl

call scan

mov [b], dl

cmp [a], dl

jl impossible

mov cx, 0

add cl, [a]

add cl, [b]

and cl, 1

jnz impossible

cwd

mov ax, 0

add al, [a]

add al, [b]

mov cx, 2

idiv cx

call print

mov dl, ' '

int 21h

cwd

mov ax, 0

mov al, [a]

sub al, [b]

mov cx, 2

idiv cx

call print

jmp newline

impossible:

lea dx, imp

mov ah, 9

int 21h

newline:

lea dx, nl

mov ah, 9

int 21h

pop cx

loop testcase

ret

main endp

scan proc

mov dx, 0

input:

mov ah, 1

int 21h

cmp al, ' '

je end

cmp al, 13

je end

push ax

mov al, 10

mul dl

mov dl, al

pop ax

sub al, '0'

add dl, al

jmp input

end:

ret

scan endp

print proc

xor cx, cx

loop1:

cwd

mov bx, 10

idiv bx

push dx

inc cx

cmp ax, 0

jg loop1

loop2:

pop dx

add dx, '0'

mov ah, 2

int 21h

loop loop2

ret

print endp

**problem : 136**

**------------------**

.model small

.stack 100h

.data

str dw "The 1500'th ugly number is 859963392$"

.code

main proc

mov ax,@data

mov ds,ax

lea dx,str

mov ah,9

int 21h

exit:

mov ah,4ch

int 21h

main endp

end main

**problem : 10929**

**-----------------------**

.model small

.stack 100h

.data

input db 1002 dup(?)

yes db ' is a multiple of 11', 10, 13, '$'

no db ' is not a multiple of 11', 10, 13, '$'

.code

main proc

mov ax, @data

mov dx, ax

testcase:

mov si, 0

mov cx, 0

mov ah, 1

int 21h

cmp al, '0'

je endcase

inputloop:

mov si[input], al

inc si

int 21h

cmp al, 13

jz endinput

jmp inputloop

endinput:

mov si[input], 36

mov cx, si

dec cx

dec cx

mov bx, 10

xor ax, ax

mov al, 0[input]

sub al, 30h

imul bx

add al, 1[input]

sub al, 30h

mov si, 2

cmp cx, 1

jl comp

division:

cwd

mov bx, 11

idiv bx

mov ax, dx

mov bx, 10

imul bx

add al, si[input]

sub al, 30h

inc si

loop division:

comp:

push ax

mov ah, 9

lea dx, input

int 21h

pop ax

mov bx, 11

cwd

idiv bx

cmp dx, 0

jz printyes

mov ah, 9

lea dx, no

int 21h

jmp testcase

printyes:

mov ah, 9

lea dx, yes

int 21h

jmp testcase

endcase:

ret

main endp

**problem : 10970**

**---------------------**

.model small

.stack 100h

.code

main proc

loop1:

;for 1st input

mov ah,1

int 21h

mov bl,al

sub bl,48

; print a space

mov ah,2

mov dl,' '

int 21h

;for 2nd input

mov ah,1

int 21h

mov bh,al

sub bh,48

and ax,0

mov al,bh

mul bl

sub ax,1

and bx,0

mov bx,ax

add bl,48

mov ah,2

mov dl,0dh

int 21h

mov dl,0ah

int 21h

mov dl,bl

int 21h

mov dl,0dh

int 21h

mov dl,0ah

int 21h

jmp loop1

mov ah,4ch

int 21h

main endp

end main

**problem : 11150**

**-----------------------**

.model small

.stack 100h

.data

n db 3

n2 db ?

.code

main proc

mov ax,@data

mov ds,ax

loop1:

; for input

mov ah,1

int 21h

mov n2,al

sub n2,48

and bl,0

mov bh,n2

loop2:

and ax,0

mov al,n2

div n

add bl,al

add al,ah

mov n2,al

cmp n2,1

jle not\_to\_add\_1

cmp n2,2

je add\_1

jmp loop2

not\_to\_add\_1:

add bl,bh

add bl,48

call newline

mov ah,2

mov dl,bl

int 21h

call newline

jmp loop1

add\_1:

add bl,bh

add bl,49

call newline

mov ah,2

mov dl,bl

int 21h

call newline

jmp loop1

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,0dh

int 21h

mov dl,0ah

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 11172**

**-----------------------**

.model small

.stack 100h

.code

main proc

mov ah,1

int 21h

mov cl,al

sub cl,48

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

loop1:

mov ah,1

int 21h

mov bl,al

mov ah,2

mov dl,' '

int 21h

mov ah,1

int 21h

mov bh,al

cmp bl,bh

jl less

jg greater

je equal

less:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

mov dl,'<'

int 21h

jmp continue\_loop

greater:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

mov dl,'>'

int 21h

jmp continue\_loop

equal:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

mov dl,'='

int 21h

jmp continue\_loop

continue\_loop:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

dec cl

cmp cl,0

jge loop1

exit:

mov ah,4ch

int 21h

main endp

end main

**problem : 11388**

**-----------------------**

.model small

.stack 100h

.data

v1 db ?

v2 db ?

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

and cx,0

mov cl,al

sub cl,48

call newline

loop1:

mov ah,1

int 21h

mov bl,al

sub bl,48

mov v1,al

mov ah,2

mov dl,' '

int 21h

mov ah,1

int 21h

mov v2,al

and dx,0

mov dl,al

sub dl,48

call newline

mov ax,dx

div bl

cmp ah,0

jne print\_negative\_1

mov ah,2

mov dl,v1

int 21h

mov dl, ' '

int 21h

mov dl,v2

int 21h

call newline

loop loop1

print\_negative\_1:

mov dl,'-'

mov ah,2

int 21h

mov dl,'1'

int 21h

call newline

loop loop1

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,13

int 21h

mov dl,10

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 11479**

**-----------------------**

.model small

.stack 100h

.data

inv db 'invalid$'

equ db 'equilateral$'

iso db 'isosceles$'

sca db 'scalene$'

nl db 10, 13, '$'

.code

main proc

mov ax, @data

mov ds, ax

call scan

mov cx, 0

mov cl, dl

testcase:

push cx

call scan

mov bh, dl

call scan

mov bl, dl

call scan

mov ch, dl

xor ax, ax

mov al, bh

add al, bl

cmp al, ch

jl invalid

xor ax, ax

mov al, bh

add al, ch

cmp al, bl

jl invalid

xor ax, ax

mov al, ch

add al, bl

cmp al, bh

jl invalid

cmp bh, bl

jne nequal

cmp bh, ch

je equilateral

jne isosceles

nequal:

cmp bh, ch

je isosceles

cmp bl, ch

je isosceles

jmp scalene

invalid:

mov ah, 9

lea dx, inv

int 21h

jmp newline

equilateral:

mov ah, 9

lea dx, equ

int 21h

jmp newline

isosceles:

mov ah, 9

lea dx, iso

int 21h

jmp newline

scalene:

mov ah, 9

lea dx, sca

int 21h

newline:

lea dx, nl

int 21h

pop cx

loop testcase

ret

main endp

scan proc

mov dx, 0

input:

mov ah, 1

int 21h

cmp al, ' '

je end

cmp al, 13

je end

push ax

mov al, 10

mul dl

mov dl, al

pop ax

sub al, '0'

add dl, al

jmp input

end:

ret

scan endp

problem : 11689

----------------

.model small

.stack 100h

.data

v db ?

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

mov cl,al

sub cl,48

call newline

test\_case:

and v,0

mov ah,1 ;

int 21h

mov bl,al

sub bl,48

;print a space

mov ah,2

mov dl,' '

int 21h

mov ah,1

int 21h

sub al,48

add bl,al

;print a space

mov ah,2

mov dl,' '

int 21h

mov ah,1

int 21h

sub al,48

mov bh,al

call newline

loop1:

and ax,0

mov al,bl

div bh

add v,al

add al,ah

mov bl,al

cmp bl,bh

jge loop1

add v,48

mov dl,v

mov ah,2

int 21h

call newline

loop test\_case

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,13

int 21h

mov dl,10

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 11854**

**-----------------------**

.model small

.stack 100h

.data

right db 'right', 10, 13, '$'

wrong db 'wrong', 10, 13, '$'

.code

main proc

start:

call scan

mov bh, dl

call scan

mov bl, dl

call scan

mov ch, dl

cmp bh, 0

jne calc

cmp bl, 0

jne calc

cmp ch, 0

je endmain

calc:

xor ax, ax

mov al, bh

imul al

mov bh, al

xor ax, ax

mov al, bl

imul al

add bh, al

xor ax, ax

mov al, ch

imul al

cmp al, bh

je printright

lea dx, wrong

mov ah, 9

int 21h

jmp start

printright:

lea dx, right

mov ah, 9

int 21h

jmp start

endmain:

ret

main endp

scan proc

mov dx, 0

input:

mov ah, 1

int 21h

cmp al, ' '

je end

cmp al, 13

je end

push ax

mov al, 10

mul dl

mov dl, al

pop ax

sub al, '0'

add dl, al

jmp input

end:

ret

scan endp

**problem : 11877**

**-----------------------**

.model small

.stack 100h

.data

n db 3

n2 db ?

.code

main proc

mov ax,@data

mov ds,ax

loop1:

; for input

mov ah,1

int 21h

mov n2,al

sub n2,48

and bl,0

cmp n2,0

je exit

loop2:

and ax,0

mov al,n2

div n

add bl,al

add al,ah

mov n2,al

cmp n2,1

jle not\_to\_add\_1

cmp n2,2

je add\_1

jmp loop2

not\_to\_add\_1:

add bl,48

call newline

mov ah,2

mov dl,bl

int 21h

call newline

jmp loop1

add\_1:

add bl,49

call newline

mov ah,2

mov dl,bl

int 21h

call newline

jmp loop1

exit:

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,13

int 21h

mov dl,10

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 11879**

**-----------------------**

.model small

.stack 100h

.data

input db 1002 dup(?)

yes db '1', 10, 13, '$'

no db '0', 10, 13, '$'

.code

main proc

mov ax, @data

mov ds, ax

testcase:

mov si, 0

mov cx, 0

mov ah, 1

int 21h

cmp al, '0'

je endcase

inputloop:

mov si[input], al

inc si

int 21h

cmp al, 13

jz endinput

jmp inputloop

endinput:

mov si[input], 36

mov cx, si

dec cx

xor ax, ax

mov al, 0[input]

sub al, 30h

mov si, 1

cmp cx, 1

jl comp

division:

mov bx, 10

imul bx

add al, si[input]

inc si

sub al, 30h

cwd

mov bx, 17

idiv bx

mov ax, dx

loop division:

comp:

cmp dx, 0

jz printyes

mov ah, 9

lea dx, no

int 21h

jmp testcase

printyes:

mov ah, 9

lea dx, yes

int 21h

jmp testcase

endcase:

ret

main endp

**problem : 12372**

**-----------------------**

.model small

.stack 100h

.data

case dw ?

a dw ?

b dw ?

c dw ?

msg db 'case $'

msg1 db 'good$'

msg2 db 'bad$'

.code

main proc

start:

mov case,0

mov ax,@data

mov ds,ax

call scan

xor cx,cx

mov cx,ax

loop1:

inc case

call scan

mov a,ax

call scan

mov b,ax

call scan

mov c,ax

cmp a,20

jle n1

jmp bad

n1:

cmp b,20

jle n2

jmp bad

n2:

cmp c,20

jle good

jmp bad

good:

mov ah,9

lea dx,msg

int 21h

mov ax,case

call print

mov ah,2

mov dx,':'

int 21h

mov dx,' '

int 21h

mov ah,9

lea dx,msg1

int 21h

mov ah,2

mov dx,10

int 21h

mov dx,13

int 21h

loop loop1

mov ah,2

mov dx,10

int 21h

mov dx,13

int 21h

jmp start

bad:

mov ah,9

lea dx,msg

int 21h

mov ax,case

call print

mov ah,2

mov dx,':'

int 21h

mov dx,' '

int 21h

mov ah,9

lea dx,msg2

int 21h

mov ah,2

mov dx,10

int 21h

mov dx,13

int 21h

loop loop1

mov ah,2

mov dx,10

int 21h

mov dx,13

int 21h

jmp start

main endp

scan proc

push bx

push cx

push dx

xor bx, bx

xor cx, cx

mov ah, 1

int 21h

cmp al, '-'

je negative

cmp al, '+'

je positive

jmp input

negative:

mov cx, 1

positive:

int 21h

input:

and ax, 000fh

push ax

mov ax, 10

mul bx

pop bx

add bx, ax

mov ah, 1

int 21h

cmp al, ' '

je endinput

cmp al, 13

je carriagereturn

jmp input

carriagereturn:

mov ah, 2

mov dl, 10

int 21h

endinput:

mov ax, bx

cmp cx, 0

je endscan

neg ax

endscan:

pop dx

pop cx

pop bx

ret

scan endp

print proc

push ax

push bx

push cx

push dx

cmp ax, 0

jge init

push ax

mov dl, '-'

mov ah, 2

int 21h

pop ax

neg ax

init:

xor cx, cx

mov bx, 10

digitify:

cwd

div bx

push dx

inc cx

cmp ax, 0

jnz digitify

mov ah, 2

printloop:

pop dx

or dl, 30h

int 21h

loop printloop

pop dx

pop cx

pop bx

pop ax

ret

print endp

**problem : 12403**

**-----------------------**

.model small

.stack 100h

.data

n db 0

.code

main proc

mov ax,@data

mov ds,ax

mov ah,1

int 21h

mov cl,al

sub cl,48

call newline

loop1:

mov bl,0

loop2:

mov ah,1

int 21h

cmp al,13

je loop2end

cmp bl,0

jne loop2

inc bl

mov bh,al

jmp loop2:

loop2end:

call newline

cmp bh,64h

jne print\_the\_result

mov ah,1

int 21h

sub al,48

add n,al

jmp continue\_loop2

print\_the\_result:

add n,48

mov ah,2

mov dl,n

int 21h

sub n,48

continue\_loop2:

call newline

dec cl

cmp cl,0

je exit

jmp loop1

exit:

mov ah,4ch

int 21h

main endp

newline proc

push ax

push dx

mov ah,2

mov dl,13

int 21h

mov dl,10

int 21h

pop dx

pop ax

ret

newline endp

end main

**problem : 12468**

**-----------------------**

.model small

.stack 100h

.code

main proc

maintop:

call scan

mov bx, ax

call scan

cmp ax, -1

jz mainend

sub ax, bx

cmp ax, 0

jg abs

neg ax

abs:

cmp ax, 50

jle printres

mov bx, ax

mov ax, 100

sub ax, bx

printres:

call print

mov ah, 2

mov dl, 10

int 21h

mov dl, 13

int 21h

jmp maintop

mainend:

ret

main endp

scan proc

push bx

push cx

push dx

xor bx, bx

xor cx, cx

mov ah, 1

int 21h

cmp al, '-'

je negative

cmp al, '+'

je positive

jmp inputscan

negative:

mov cx, 1

positive:

int 21h

inputscan:

and ax, 000fh

push ax

mov ax, 10

mul bx

pop bx

add bx, ax

mov ah, 1

int 21h

cmp al, ' '

je endinput

cmp al, -1

je endinput

cmp al, 13

je carriagereturn

jmp inputscan

carriagereturn:

mov ah, 2

mov dl, 10

int 21h

endinput:

mov ax, bx

cmp cx, 0

je endscan

neg ax

endscan:

pop dx

pop cx

pop bx

ret

scan endp

print proc

push ax

push bx

push cx

push dx

cmp ax, 0

jge init

push ax

mov dl, '-'

mov ah, 2

int 21h

pop ax

neg ax

init:

xor cx, cx

mov bx, 10

digitify:

cwd

div bx

push dx

inc cx

cmp ax, 0

jnz digitify

mov ah, 2

printloop:

pop dx

or dl, 30h

int 21h

loop printloop

pop dx

pop cx

pop bx

pop ax

ret

print endp

**problem : 12577**

**-----------------------**

.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

huddai:

int 21h

cmp al, 13

jne huddai

cmp bl, '\*'

je exit

cmp bl, 'h'

je akbar

lea dx, umrah

mov ah, 9

int 21h

jmp start

akbar:

lea dx, hajj

mov ah, 9

int 21h

jmp start

exit:

ret

main endp

**problem : 12646**

**-----------------------**

.model small

.stack 100h

.code

main proc

start:

mov ah, 1

int 21h

mov bh, al

mov ah, 2

mov dl, ' '

int 21h

mov ah, 1

int 21h

mov bl, al

mov ah, 2

mov dl, ' '

int 21h

mov ah, 1

int 21h

mov ch, al

mov ah, 2

mov dl, ' '

int 21h

cmp bh, bl

jne nequal

cmp bh, ch

je none

mov ah, 2

mov dl, 'c'

int 21h

jmp newline

nequal:

cmp bh, ch

jne nequal2

mov ah, 2

mov dl, 'b'

int 21h

jmp newline

nequal2:

mov ah, 2

mov dl, 'a'

int 21h

jmp newline

none:

mov ah, 2

mov dl, '\*'

int 21h

newline:

mov ah, 2

mov dl, 10

int 21h

mov dl, 13

int 21h

jmp start

endmain:

ret

main endp