; Tinh n!, n<7

.Model small

.Stack 100H

.Data

; source string

str1 DB 'Nhap mot so: $'

str2 DB 13, 10, 'Giai thua la: $'

str3 DB 13, 10, 'Invalid number entered!$'

.code

main proc

; initilize the ds and es registers

mov ax, @Data

mov ds,ax

; print the prompt

lea dx, str1

call printString

; get the number from keyboard

mov cl, 0

mov bl, 10

READ:

mov ah, 1

int 21h

cmp al, 13

je ENDREAD

cmp al, '0'

jl ERROR

cmp al, '9'

jg ERROR

sub al, '0'

mov dl, al

mov dh, 0

mov al, cl

mul bl

add ax, dx

mov cl, al

jmp READ

ENDREAD:

;Calculate n!

xor ch, ch

mov ax, 1

GT:

mul cx

dec cx

je GTEND

jmp GT

GTEND:

; print the number

lea dx, str2

call printString

call printNumber

jmp THEEND

ERROR:

; print the error message

lea dx, str3

call printString

THEEND:

; end program

mov ah, 4CH

int 21H

main endp

;----------------------------------------

; print an integer number (MAX: 2550)

; input: AX: input number to print

printNumber proc

push ax

push bx

push cx

push dx

; check if AX is negative --> change to positive and print the minus sign (-)

cmp ax, 0

jge NOT\_NEGATIVE

; change to positive

neg ax

; print the minus sign

mov dl, '-'

call printCharacter

; divide the number by 10 to get the remainder and the quotient

NOT\_NEGATIVE:

mov bl, 10

mov cx, 0

StartSplit:

div bl

push ax ; store the result into stack

inc cx ; count the number of digit

cmp al, 0

jz ExitSplit

xor ah, ah

jmp StartSplit

ExitSplit:

; print each digit

StartPrint:

pop ax

mov dl, ah

call printSingleDigit

loop StartPrint

pop dx

pop cx

pop bx

pop ax

ret

printNumber endp

;----------------------------------------

; proc to print out a single digit number

; input: dl to contain the digit to print

printSingleDigit proc

push ax

add dl, '0' ; digit to char

mov ah, 2

int 21H

pop ax

ret

printSingleDigit endp

; --------------------------------------

; proc to print a string

; input: DX to contain the relative address of the string

printString proc

push ax ; store AX into stack

mov ah, 9

int 21H

pop ax ; restore AX from stack

ret

printString endp

;----------------------------------------

; proc to print out a character

; input: dl to contain the character to print

printCharacter proc

push ax

mov ah, 2

int 21H

pop ax

ret

printCharacter endp

end main

; Print Enlish and Vietnamese greetings

.Model Small

.Stack 100

.Data

CRLF DB 13, 10, '$'

ChaoTay DB 'hello!$'

ChaoTa DB 'Chao ban!$'

.Code

MAIN Proc

MOV AX, @Data ; khoi dau thanh ghi DS

MOV DS, AX

; hien thi loi chào dùng hàm 9 c?a INT 21H

MOV AH, 9

LEA DX, ChaoTay

INT 21H

; cách 5 dòng dùng hàm 9 cua INT 21H

LEA DX, CRLF

MOV CX, 6 ;CX chua so dòng cách +1

LAP: INT 21H

LOOP LAP

; hien thi loi chào dùng hàm 9 c?a INT 21H

LEA DX, ChaoTa

INT 21H

; tro ve DOS dùng hàm 4 CH cua INT 21H

MOV AH, 4CH

INT 21H

MAIN Endp

END MAIN

; convert lower-case chars in a string to upper-case

' print 2 strings

.Model small

.Stack 100H

.Data

; source string

str1 DB 'a','5', 'B', '?', 'd', 'g', 'P','N','k','\*'

DB 10,13,'$'

; destination string

str2 DB 10 DUP(' ')

DB '$'

.code

main proc

; initilize the ds and es registers

mov ax, @Data

mov ds,ax

mov es,ax

; make SI points to str1 and DI to str2

lea si, str1

lea di, str2

cld

mov cx, 10

Start:

lodsb

; check if it is lower case

cmp al, 'a'

jl NotLowerCase

cmp al, 'z'

jg NotLowerCase

; is lower case, convert to upper case

sub al, 20H

; store to new string

NotLowerCase: stosb

loop Start

; print the original string

lea dx, str1

mov ah, 9

int 21H

; print the output

lea dx, str2

mov ah, 9

int 21H

; end program

mov ah, 4CH

int 21H

main endp

end main

; Find max value in an array and print out it

.Model small

.Stack 100H

.Data

; source string

list DB 1,4,0,9,7,2,4,6,2,5

.code

main proc

; initilize the ds and es registers

mov ax, @Data

mov ds,ax

cld

mov cx, 9

lea si, list ; si points to list

mov bl, [si] ; max <-- 1st element

inc si

Start:

lodsb

cmp al, bl

jle BYPASS

mov bl, al; al>bl --> bl to store new max

BYPASS:

loop Start

; print the max

add bl, '0' ; digit to char

mov dl,bl

mov ah, 2

int 21H

; end program

mov ah, 4CH

int 21H

main endp

End Main

; Tinh tong 1 mang cac so chia het cho 7

; In KQ duoi dang so hexa

.Model small

.Stack 100H

.Data

; source string

str1 DB 'Tong theo hexa la : $'

mang dw 10, 14, 7, 0, 9, 20, 34, 12, 35, 11

.code

main proc

; initilize the ds and es registers

mov ax, @Data

mov ds,ax

mov bx, 0

mov dl, 7 ; so chia de kiem tra

mov cx, 10

lea si, mang

SUM:

lodsw

push ax

div dl

cmp ah, 0

jne SKIP

pop ax

add bx, ax ; Cong neu chia het cho 7

SKIP:

loop SUM

; In ket qua

lea dx, str1

call printString

mov ax, bx

call printNumberHex

; end program

mov ah, 4CH

int 21H

main endp

;----------------------------------------

; print an integer number (MAX: 2550)

; input: AX: input number to print

printNumberHex proc

push ax

push bx

push cx

push dx

; divide the number by 16 to get the remainder and the quotient

mov bl, 16

mov cx, 0

StartSplit:

div bl

push ax ; store the result into stack

inc cx ; count the number of digit

cmp al, 0

jz ExitSplit

xor ah, ah

jmp StartSplit

ExitSplit:

; print each digit

StartPrint:

pop ax

mov dl, ah

call printSingleDigit

loop StartPrint

pop dx

pop cx

pop bx

pop ax

ret

printNumberHex endp

;----------------------------------------

; proc to print out a single digit hex number

; input: dl to contain the digit to print

printSingleDigit proc

push ax

cmp dl, 9

jg HEX

add dl, '0' ; digit to char

jmp PRINT

HEX:

cmp dl, 10

je HEX\_A

cmp dl, 11

je HEX\_B

cmp dl, 12

je HEX\_C

cmp dl, 13

je HEX\_D

cmp dl, 14

je HEX\_E

mov dl, 'F'

jmp PRINT

HEX\_A: mov dl, 'A'

jmp PRINT

HEX\_B: mov dl, 'B'

jmp PRINT

HEX\_C: mov dl, 'C'

jmp PRINT

HEX\_D: mov dl, 'D'

jmp PRINT

HEX\_E: mov dl, 'E'

PRINT:

mov ah, 2

int 21H

pop ax

ret

printSingleDigit endp

; --------------------------------------

; proc to print a string

; input: DX to contain the relative address of the string

printString proc

push ax ; store AX into stack

mov ah, 9

int 21H

pop ax ; restore AX from stack

ret

printString endp

end main

; Tim USCLN va BSCNN cua 2 so

; In ket qua ra man hinh

.Model small

.Stack 100H

.Data

; source string

str1 DB 'So thu nhat la: $'

str2 DB 13, 10, 'So thu hai la: $'

str3 DB 13, 10, 'USCLN la: $'

str4 DB 13, 10, 'BSCNN la: $'

.code

main proc

; initilize the ds and es registers

mov ax, @Data

mov ds,ax

mov cl, 26 ; so thu nhat

mov ch, 8 ; so thu hai

; tim USCLN bang pp tru dan

mov dl, cl

mov dh, ch

USC:

cmp dl, dh

je USCEND

jg SOLN

sub dh, dl ; b= b-a

jmp USC

SOLN: sub dl, dh ; a = a-b

jmp USC

USCEND:

push dx ; luu kq USCLN vao stack

; In 2 so ban dau

lea dx, str1

call printString

xor ah, ah

mov al, cl

call printNumber

lea dx, str2

call printString

xor ah, ah

mov al, ch

call printNumber

; In USCLN

lea dx, str3

call printString

pop dx

xor ah, ah

mov al, dl

call printNumber

; Tim BSCNN bang pp quet tu max(a, b) den a\*b

mov al, cl

mul ch ; AX = a\*b

mov dl, cl

cmp cl, ch

jle BSC

mov dl, ch

xor dh, dh ; DX = max(a, b)

mov bx, ax ; BX = a\*b

BSC:

mov ax, dx

div cl

cmp ah, 0

jne TIEPTUC

mov ax, dx

div ch

cmp ah, 0

jne TIEPTUC

jmp BSCEND

TIEPTUC:

inc dx

cmp dx, bx

jg BSCEND

jmp BSC

BSCEND:

push dx

; in BSCNN

lea dx, str4

call printString

pop dx

mov ax, dx

call printNumber

; end program

mov ah, 4CH

int 21H

main endp

;----------------------------------------

; print an integer number (MAX: 2550)

; input: AX: input number to print

printNumber proc

push ax

push bx

push cx

push dx

; check if AX is negative --> change to positive and print the minus sign (-)

cmp ax, 0

jge NOT\_NEGATIVE

; change to positive

neg ax

; print the minus sign

mov dl, '-'

call printCharacter

; divide the number by 10 to get the remainder and the quotient

NOT\_NEGATIVE:

mov bl, 10

mov cx, 0

StartSplit:

div bl

push ax ; store the result into stack

inc cx ; count the number of digit

cmp al, 0

jz ExitSplit

xor ah, ah

jmp StartSplit

ExitSplit:

; print each digit

StartPrint:

pop ax

mov dl, ah

call printSingleDigit

loop StartPrint

pop dx

pop cx

pop bx

pop ax

ret

printNumber endp

;----------------------------------------

; proc to print out a single digit number

; input: dl to contain the digit to print

printSingleDigit proc

push ax

add dl, '0' ; digit to char

mov ah, 2

int 21H

pop ax

ret

printSingleDigit endp

; --------------------------------------

; proc to print a string

; input: DX to contain the relative address of the string

printString proc

push ax ; store AX into stack

mov ah, 9

int 21H

pop ax ; restore AX from stack

ret

printString endp

;----------------------------------------

; proc to print out a character

; input: dl to contain the character to print

printCharacter proc

push ax

mov ah, 2

int 21H

pop ax

ret

printCharacter endp

end main

.MODEL small

.STACK

.DATA

tb1 DB 'Nhap vao 1 chuoi: $'

tb2 DB 10,13,'Doi thanh chu thuong: $'

tb3 DB 10,13,'Doi thanh chu hoa: $'

s DB 100,?,101 dup('$')

.CODE

BEGIN:

MOV AX, @DATA

MOV DS,AX

;xuat chuoi tb1

MOV AH,09h

LEA DX,tb1

INT 21h

;nhap chuoi s

MOV AH,0AH

LEA DX,s

INT 21h

;xuat chuoi tb2

MOV AH,09h

LEA DX,tb2

INT 21h

; Goi chuong trinh con in chuoi thuong

CALL InChuoiThuong

; xuat chuoi tb3

MOV AH,09h

LEA DX,tb3

INT 21h

; Goi chuong trinh con in chuoi thuong

CALL InChuoiHoa

MOV AH,4ch

INT 21h

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Doi thanh chuoi ky tu thuong

InChuoiThuong PROC

LEA SI,s+1

XOR CX,CX

MOV CL,[SI]

INC SI

LapThuong:

MOV AH,02h

MOV DL,[SI]

CMP DL,'A'

JB LT1

CMP DL,'Z'

JA LT1

ADD DL,32

LT1: INC SI

INT 21h

LOOP LapThuong

RET

InChuoiThuong ENDP

; Doi thanh chuoi ky tu hoa

InChuoiHoa PROC

LEA SI,s+1

XOR CX,CX

MOV CL,[SI]

INC SI

LapHoa:

MOV AH,02h

MOV DL,[SI]

CMP DL,'a'

JB LH1

CMP DL,'z'

JA LH1

SUB DL,32

LH1: INC SI

INT 21h

LOOP LapHoa

RET

InChuoiHoa ENDP

END BEGIN