;LEE JUN LAM TP055697

.model small

.stack 100h

.386 ;Add 386 directory prevent jump out of range

.data

MP db "Main Page $" ; DISPLAY MESSAGE

MP1 db 10,13,10,13, "Input 1 for Number Pattern (Diamond) $"

MP2 db 10,13,10,13, "Input 2 for Design Pattern (Wave) $"

MP3 db 10,13,10,13, "Input 3 for Box Type Pattern (Square)$"

MP4 db 10,13,10,13, "Input 4 for Nested Loop Pattern (Triangle)$"

MP5 db 10,13,10,13, "Input 5 for Special Pattern $" ; 10 is the ASCII control code for line feed

INPUT db 10,13,10,13, "Input (1, 2, 3, 4, 5, 0 (exit)): $" ; while 13 is the code for carriage return

ERROR db 10,13,10,13, "Invalid Input! Try Again! (Enter Any Key to continue) $"

ALERT db 10,13,10,13, "Run The Program Again? Yes/No: $"

choice db ?

; Diamond Pattern

number db ?

space db ?

; Wave Pattern

sec1 dw ?

sec2 dw ?

sec4 dw ?

sec3 dw ?

sec5 dw ?

sec6 dw ?

;Box Pattern

length db ? ; 9 rows

width db ? ; 9 columns

;Triangle Pattern

flag db ?

current db ?

spacing db ? ; 1st line space 5 times (6666-----9999)

character db ? ; 1st line display 4 character of 6 + 4 character of 9

;Cross Pattern

hw db "\*",'$'

col1 db ?

row1 db ?

col2 db ?

;-----------------------------------------CODING BEGIN-----------------------------------------------------

.code

Main proc

;------------------------------------------MAIN PAGE-------------------------------------------------------

Main: ; DISPLAY MAIN PAGE INTERFACE

mov ax, @data

mov ds, ax

mov ah, 9

mov dx, offset MP ; 9 display a string of characters whose offset is specified by DX

int 21h

mov ah, 9

mov dx, offset MP1

int 21h

mov ah, 9

mov dx, offset MP2

int 21h

mov ah, 9

mov dx, offset MP3

int 21h

mov ah, 9

mov dx, offset MP4

int 21h

mov ah, 9

mov dx, offset MP5

int 21h

mov ah, 9

mov dx, offset INPUT

int 21h

mov ah, 1

int 21h

mov choice, al

cmp choice, 00h

jbe InvalidInput

cmp choice, 36h

ja InvalidInput

cmp choice, 30h

je close

cmp choice, 31h

je DiamondPattern

cmp choice, 32h

je WavePattern

cmp choice, 33h

je BoxPattern

cmp choice, 34h

je RightAngleTriangles

cmp choice, 35h

je Cross

InvalidInput: ; INVALID INPUT DURING MAIN PAGE

mov ah, 9

mov dx, offset ERROR

int 21h

mov ah, 1

int 21h

mov choice, al

cmp choice, 'y'

je MAIN

Final:

mov dx, offset ALERT

mov ah, 9

int 21h

mov ah, 7

int 21h

mov choice, al

cmp choice, 'Y'

mov al,2

mov ah,0

int 10h

je Main

cmp choice, 'y'

mov al,2

mov ah,0

int 10h

je Main

cmp choice, 'N'

mov al,2

mov ah,0

int 10h

je close

cmp choice, 'n'

mov al,2

mov ah,0

int 10h

je close

jnp InvalidInput

;-----------------------------------------Number Pattern (Diamond Shape)-----------------------------------

DiamondPattern:

mov ah, 2

mov dl, 10 ;ascii ---> 10 New Line

int 21h

;-------------Diamond Upper Part--------------

mov cx,5 ;Step 1: Create 5 rows of line for upper triangle (cx = count register)

mov bx,1 ;Assign 1 to memory variable bx (In this case we start from 1 number in 1st line)

part1:

push cx ;The push instruction places its operand (1st iteration cx = 5, 2nd iteration cx = 4)

part2:

mov ah,2

mov dl,32 ;ascii ---> 32 Space (space 5 times in first line)

int 21h

loop part2

mov cx,bx ;moves bx address to cx (when cx=first iteration, bx = 1 ; when cx is second loop, bx = 1+1+1)

part3:

mov ah,2

mov dl,56 ;ascii ---> 56 Display character 8 (first line is bx = 1, so will display one 8)

int 21h

loop part3

; now the first line is complete, proceed to the next line

mov ah,2

mov dl,10 ;ascii ---> 10 New Line

int 21h

mov dl,13 ;ascii ---> 13 Carriage Return

int 21h

; prepare to increment for next 4 lines

inc bx ; + 1 number for the next line

inc bx ; + 1 number for the next line = + 2 numbers for the next line

pop cx ; Proceed to the next iteration/ next line

loop part1 ; Back to part1 for another 4 lines (until 5th line is completed)

;--------------Diamond Lower Part---------------

mov cx,4 ; Step 2: Print the lower triangle (4 rows)

mov bh,7 ; Start from 7 numbers (\*\*\*\*\*\*\*) to 1 number (\*)

mov bl,2 ; Assign 2 to bl register

mov number,bh ; assign bh (7) to number

mov space,bl ; assign bl (2) to space

part4:

cmp space,0 ; cmp is compare. If space = 0

je part5 ; then jump to part5

mov ah,2 ; If space != 0 then continue proceed

mov dl,32 ; ascii ---> 32 Space (first line bl = 2 so space 2 times, second line bl = 3 so space 3 )

int 21h

dec space ; decrement (first line bl 2-1-1 = thus space 2 times)

jmp part4 ; and loop back to part4

part5:

mov ah,2

mov dl,'8' ; display 8

int 21h

dec number ; decrement character (7-1 until all 7 number display in 1st line of lower triangle)

cmp number,0 ; compare if character = 0

jne part5 ; then jump back to part5 if NOT equal to continue display the number 8

part6:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

dec bh ; decrement bh (-1)

dec bh ; decrement bh (-1) -1-1=-2

mov number,bh ; move bh address to character (7-2 =5, move the 5 to the number, the next line display 5 number)

inc bl ; increment bl (2+1 = 3)

mov space,bl ; move bl (3) to space

loop part4

jmp Final

;-----------------------------------------Design Pattern (Wave)--------------------------------------------

WavePattern:

mov sec1,0

mov sec2,0

mov sec3,3

mov sec4,7

mov sec5,0

mov sec6,1

mov ah, 2

mov dl, 10 ;ascii ---> 10 New Line

int 21h

Begin:

inc sec1 ; sec1 = 0 + 1 =1

cmp sec1,2 ; if sec1 = 2

jl restart ; jump to restart if sec1 < 2

mov cx,sec2 ; assign sec2 (0) to cx -> cx = 0

loop1: ; begin loop funtion

cmp sec2,0 ; if sec2 = 0

je display3 ; then go to display3

mov dl,32 ; else spacing

mov ah,2

int 21h

loop loop1 ; continue des\_loop spacing

display3:

inc sec2 ; increment sec2 (0 + 1 =1)

cmp sec1,5 ; if sec1 = 5

je restart ; then go the restart loop

mov dl,51

mov ah,2

int 21h

restart:

mov cx,sec3 ; assign c3 (3) to cx register (cx = 3)

designstartingpattern :

mov bx,cx ; asign cx address to bx register (when cx =3, bx =3)

mov cx,sec4 ; assign sec4 (7) to cx register

cmp sec1,5 ; compare sec1 = 5

je display0 ; if sec1 = 5 then jump to display0 ----------else go to spacing1

spacing1: ; else spacing 7 times first (for the first line)

mov ah,2

mov dl,32

int 21h

loop spacing1

display0: ; then display 0 (on the first line)

mov ah,2

mov dl,48

int 21h

mov cx,sec5 ; assign sec5 (0) to cx

spacing2:

cmp sec5,0

je display4 ;if sec5 = 0 then proceed to display4 to display 4 immediately(for first line)

mov ah,2

mov dl,32 ; else spacing base on cx value

int 21h

loop spacing2

display4:

cmp sec1,5 ; if sec1 = 5 then proceed to space\_jump

mov ah,2

je space\_jump

mov dl, 52

int 21h

space\_jump:

mov cx,bx

loop designstartingpattern

cmp sec1,5

jne lowerhalf

mov ah,2

mov dl,48

int 21h

lowerhalf :

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

cmp sec6,5

je finishing

inc sec5

inc sec5

dec sec4

dec sec4

inc sec6

cmp sec6,6

jne begin

finishing :

dec sec2

dec sec2

inc sec4

inc sec4

dec sec5

dec sec5

startingpattern1:

dec sec1

cmp sec1,2

jl des\_loop3

mov cx,sec2

des\_loop1:

cmp sec2,0

je des\_loop2

mov ah,2

mov dl,32

int 21h

loop des\_loop1

des\_loop2:

dec sec2

mov ah,2

mov dl,49

int 21h

des\_loop3:

mov cx,sec3

endingofdes:

mov bx,cx

mov cx,sec4

spacing3:

mov ah,2

mov dl,32

int 21h

loop spacing3

display2:

mov ah,2 ; display character 2

mov dl,50

int 21h

mov cx,sec5

spacing4:

cmp sec5,0

je display1

mov ah,2

mov dl,32

int 21h

loop spacing4

display1:

mov ah,2

mov dl,49 ; display character 1

int 21h

mov cx,bx

loop endingofdes

finalendingofdes:

mov ah,2

mov dl,10

int 21h

mov dl,13

int 21h

dec sec5

dec sec5

inc sec4

inc sec4

dec sec6

cmp sec6,1

jne startingpattern1

jmp Final

;---------------------------------------Box Type Pattern (Square)------------------------------------------

BoxPattern:

mov dl, 10 ;ascii ---> 10 New Line

mov ah, 2

int 21h

mov length, 9

mov width, 9

mov cl, length ; assign length (9 rows) to cl register

L:

mov length, cl

mov cl, width

W:

mov ah, 2

mov dl, '0'

int 21h

mov ah, 2

mov dl, 32

int 21h

loop W

mov dl, 10

mov ah, 2

int 21h

mov dl, 13

int 21h

mov cl, length

loop L

jmp Final

;---------------------------------------Nested Loop Pattern (Triangle)--------------------------------------

RightAngleTriangles:

mov dl, 10 ;ascii ---> 10 New Line

mov ah, 2

int 21h

mov character, 4

mov spacing, 5

mov flag,0

mov ch, 0 ; assign 0 in ch register

mov al, character ; assign character (4) to al register

mov cl, 2 ; assign 2 to cl register (2 count)

mul cl ; mul means multiply (2x2 = 4)

dec al ; decrementing an operand by one by one (4-1-1-1 until 0)

mov cl, al ; move al address to cl register (when cl =1, al =4 when cl=2, al = 3)

New:

mov current, cl ; assign cl address to current

mov cl, character ; assign character (4) to cl register (in the first line, we display 4 character 6666)

L\_Right\_Angle\_Triangle:

mov ah, 2

mov dl, '6' ; display character 6 and loop (1st line display 4 times of 6, 2nd line display 3 times of 6)

int 21h

loop L\_Right\_Angle\_Triangle

mov cl, spacing ; now assign spacing (5) to cl register

lineSpaces:

mov dl, 32 ; space 5 times (6666\_\_\_\_\_9999)

mov ah, 2

int 21h

loop lineSpaces

mov cl, character ; once again, assign character (4) to cl register (we will use to display 9)

R\_Right\_Angle\_Triangle:

mov dl, '9' ; display character 9 and loop until four 9 display (for first line)

mov ah, 2

int 21h

loop R\_Right\_Angle\_Triangle

mov ah, 2

mov dl, 10 ; proceed to the next line

int 21h

mov dl, 13

int 21h

cmp character, 1 ; Compares two operands. If character equals to 1

jne lowertri ; If NOT equals to 1 the jump to lowertri loop

mov flag, 1 ; assign 1 to flag then proceed to lowertri

lowertri: ; NOTE: THIS IS FOR LOWER PART TRIANGLE ONLY

cmp flag, 1 ; compare when flag = 1

jne uppertri ; If NOT equals to 1 then jump to uppertri loop 66 \_\_\_\_\_\_\_ 99

inc character ; increment character by 1 (1+1=2) 666 \_\_\_\_\_ 999

sub spacing, 2 ; spacing subtract by 2 (5-2=3) 6666\_\_\_9999

jmp nextTriIteration ; then jump to nextTriIteration

uppertri: ; NOTE: THIS IS FOR UPPER PART TRIANGLE ONLY 6666\_\_\_\_\_9999

dec character ; character(4-1=3) 666 \_\_\_\_\_\_\_ 999

add spacing, 2 ; spacing(5+2=7) 66 \_\_\_\_\_\_\_\_\_ 99

jmp nextTriIteration ; 6 \_\_\_\_\_\_\_\_\_\_\_ 9

nextTriIteration:

mov ch, 0

mov cl, current

loop New

jmp Final

;---------------------------------------Special Pattern (Cross)--------------------------------------

cross:

mov ah, 2

mov dl, 10 ;ascii ---> 10 new line

int 21h

mov col1, 0

mov row1, 13

mov col2, 8

mov cx, 9 ;number of iterations (9 stars on each side)

cro:

; set cursor position (for left side cross)

mov ah, 2

mov bx,0

mov dl, col1 ;column

mov dh, row1 ; the row is use to move 13 position down (during screen display)

int 10h

; print hw (\*) - for the left side cross

mov ah,09h ;

mov dx, offset hw

int 21h

; set cursor position (for right side cross)

mov ah, 2

mov dl, col2 ; move cursor 8 position forward

mov dh, row1 ; the row is use to move 13 position down (during screen display)

int 10h

mov dx, offset hw ; print hw (\*) - for the right side cross

mov ah,09h

int 21h

add row1,1

add col1,1

sub col2,1

loop cro

jmp final

close:

mov ah, 4ch ; Terminate Application

int 21h

main endp

end main