

```
.386p
sseg      segment stack use16 'STACK'
          db 64 dup(?)
sseg      ends

dseg      segment para public use16 'DATA'

tab       label byte
          db 59
          dw offset cseg: flPressed
          db 71
          dw offset cseg: homePressed
          db 72
          dw offset cseg: upPressed
          db 73
          dw offset cseg: pageUpPressed
          db 75
          dw offset cseg: leftPressed
          db 77
          dw offset cseg: rightPressed
          db 79
          dw offset cseg: endPressed
          db 80
          dw offset cseg: downPressed
          db 81
          dw offset cseg: pageDownPressed
          db 0

flStr     db 'fl$'
homeStr   db 'home$'
upStr     db 'up$'
pUpStr    db 'page up$'
leftStr   db 'left$'
rghtStr   db 'right$'
endStr    db 'end$'
downStr   db 'down$'
pDwnStr   db 'page down$'

dseg      ends

cseg      segment para public use16 'CODE'
          assume cs:cseg, ds:dseg, ss:sseg

main      proc far
          push ds
          push 0
          mov ax, dseg
          mov ds, ax
          call input
          ret
main      endp

input     proc near

          push ax
          push bx
          push cx
          push dx
          mov ah, 15
          int 10h
          mov ah, 0
          mov al, 2
          int 10h

top:
          mov dx, 0
          mov al, 201
          mov cx, 1
          call print
          mov dl, 1
          mov al, 205
          mov cx, 78
          call print
          mov dl, 79
          mov al, 187
          mov cx, 1
          call print

          mov dh, 1
          mov al, 186
          mov cx, 1

sides:
          mov dl, 0
          call print
          mov dl, 79
          call print
          inc dh
          cmp dh, 24
          jl  sides

bottom:
          mov dh, 24
          mov dl, 0
          mov al, 200
          mov cx, 1
          call print
          mov dl, 1
          mov al, 205
          mov cx, 78
          call print
          mov dl, 79
          mov al, 188
          mov cx, 1
          call print

          mov dx, 0101h
          mov ah, 2
          int 10h

inputLoop:

          mov ah, 0
          int 16h
          cmp al, 0
          je key
          cmp al, 113
          je return
          call colorPrint
          jmp inputLoop

key:
          lea bx, tab
keyLoop:
          cmp byte ptr[bx], 0
          je unknownButton
          cmp ah, [bx]
          je printButton
          add bx, 3
          jmp keyLoop

printButton:
          inc bx
          call word ptr[bx]
          jmp inputLoop

unknownButton:
          mov al, 63
          call colorPrint
          jmp inputLoop

return:
          pop dx
          pop cx
          pop bx
          pop ax
          ret
input     endp

print     proc near
          push bx
          push cx

          jmp print_

printNewLine:
          inc dh
          mov dl, 1
          mov ah, 2
          int 10h

print_:
          mov ah, 2
          int 10h
          mov ah, 10
          int 10h
          inc dl
          cmp dl, 79
          je printNewLine

          mov ah, 2
          int 10h

          pop cx
          pop bx

          ret
print     endp

colorPrint proc near
          push bx
          jmp colorPrint_

colorPrintNewLine:
          inc dh
          mov dl, 1
          mov ah, 2
          int 10h
          cmp dh, 24
          je colorPrintResetCursor
          jmp colorPrintExit

colorPrintResetCursor:
          mov dx, 0101h
          mov ah, 2
          int 10h
          jmp colorPrintExit

colorPrint_:
          mov ah, 2
          int 10h
          mov bl, 5fh
          mov ah, 9
          int 10h
          inc dl
          cmp dl, 79
          je colorPrintNewLine

          mov ah, 2
          int 10h

colorPrintExit:
          pop bx
          ret
colorPrint endp

flPressed proc near
          lea bx, flStr
flPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne flPressedLoop

          ret
flPressed endp

homePressed proc near
          lea bx, homeStr
homePressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne homePressedLoop

          ret
homePressed endp

upPressed proc near
          lea bx, upStr
upPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne upPressedLoop

          ret
upPressed endp

pageUpPressed proc near
          lea bx, pUpStr
pageUpPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne pageUpPressedLoop

          ret
pageUpPressed endp

leftPressed proc near
          lea bx, leftStr
leftPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne leftPressedLoop

          ret
leftPressed endp

rightPressed proc near
          lea bx, rghtStr
rightPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne rightPressedLoop

          ret
rightPressed endp

endPressed proc near
          lea bx, endStr
endPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne endPressedLoop

          ret
endPressed endp

downPressed proc near
          lea bx, downStr
downPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne downPressedLoop

          ret
downPressed endp

pageDownPressed proc near
          lea bx, pDwnStr
pageDownPressedLoop:
          mov al, byte ptr [bx]
          call colorPrint
          inc bx
          cmp byte ptr [bx], 36
          jne pageDownPressedLoop

          ret
pageDownPressed endp

cseg      ends

          end main
```