Gregory Pierot

CPSC 240

// Gregory Pierot

// Project 4

//Question 1

#include <iostream>

using namespace std;

int main()

{

\_asm

{

printer:

mov eax,4

mov ebx,1

mov ecx,outputMsg1

mov edx,lenoutputMsg1

int 80h

mov ax,[num]

mov bx,1100000000000000b

and bx,ax

shr bx,0ch

cmp bx,00h

je printer0

cmp bx,01h

je printer1

cmp bx,02h

je printer2

jmp printer3

printer0:

mov eax,4

mov ebx,1

mov ecx,pt0

mov edx,lenpt0

int 80h

jmp floppy

printer1:

mov eax,4

mov ebx,1

mov ecx,pt1

mov edx,lenpt1

int 80h

jmp floppy

printer2:

mov eax,4

mov ebx,1

mov ecx,pt2

mov edx,lenpt2

int 80h

jmp floppy

printer3:

mov eax,4

mov ebx,1

mov ecx,pt3

mov edx,lenpt3

int 80h

floppy:

mov eax,4

mov ebx,1

mov ecx,outputMsg2

mov edx,lenoutputMsg2

int 80h

mov ax,[num]

mov bx,0000000011000000b

and bx,ax

shr bx,05h

cmp bx,00h

je floppy0

cmp bx,01h

je floppy1

cmp bx,02h

je floppy2

jmp floppy3

floppy0:

mov eax,4

mov ebx,1

mov ecx,fp0

mov edx,lenfp0

int 80h

jmp ram

floppy1:

mov eax,4

mov ebx,1

mov ecx,fp1

mov edx,lenfp1

int 80h

jmp ram

floppy2:

mov eax,4

mov ebx,1

mov ecx,fp2

mov edx,lenfp2

int 80h

jmp ram

floppy3:

mov eax,4

mov ebx,1

mov ecx,fp3

mov edx,lenfp3

int 80h

ram:

mov eax,4

mov ebx,1

mov ecx,outputMsg3

mov edx,lenoutputMsg3

int 80h

mov ax,[num]

mov bx,0000000000001100b

and bx,ax

shr bx,02h

cmp bx,00h

je ram1

cmp bx,01h

je ram2

cmp bx,02h

je ram3

jmp ram4

ram1:

mov eax,4

mov ebx,1

mov ecx,ram1m

mov edx,lenram1m

int 80h

jmp \_exit

ram2:

mov eax,4

mov ebx,1

mov ecx,ram2m

mov edx,lenram2m

int 80h

jmp \_exit

ram3:

mov eax,4

mov ebx,1

mov ecx,ram3m

mov edx,lenram3m

int 80h

jmp \_exit

ram4:

mov eax,4

mov ebx,1

mov ecx,ram4m

mov edx,lenram4m

int 80h

\_exit:

mov eax,1

mov ebx,0

int 80h

num dw 110011101001100b

outputMsg1 db 'The number of printers connected to the computer are :'

lenoutputMsg1 equ $-outputMsg1

outputMsg2 db 'The number of floppy drives are :'

lenoutputMsg2 equ $-outputMsg2

outputMsg3 db 'The size of RAM is :'

lenoutputMsg3 equ $-outputMsg3

ram1m db '16GB RAM'

lenram1m equ $-ram1m

ram2m db '32GB RAM'

lenram2m equ $-ram2m

ram3m db '48GB RAM'

lenram3m equ $-ram3m

ram4m db '64GB RAM'

lenram4m equ $-ram4m

pt0 db '0 printer'

lenpt0 equ $-pt0

pt1 db '1 printer'

lenpt1 equ $-pt1

pt2 db '2 printer'

lenpt2 equ $-pt2

pt3 db '3 printer'

lenpt3 equ $-pt3

fp0 db '1 floppy'

lenfp0 equ $-fp0

fp1 db '2 floppy'

lenfp1 equ $-fp1

fp2 db '3 floppy'

lenfp2 equ $-fp2

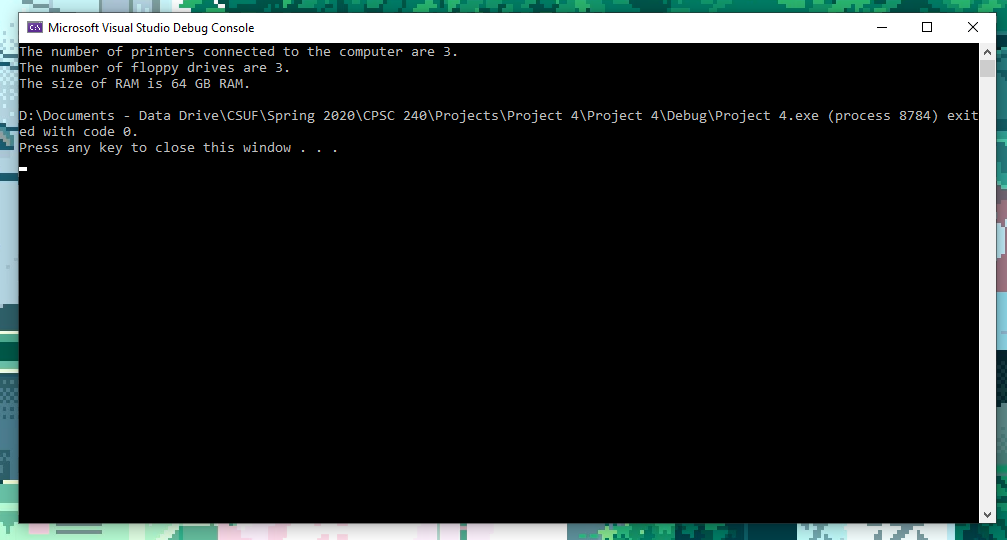
fp3 db '4 floppy'

lenfp3 equ $-fp3

}

return 0;

}



// Question 2

\_asm

{

; i) Print "Ax="

mov edx, OFFSET msg1

call WriteString

; Display register AX

mov ax, 0x6a2f

mov ebx, TYPE 2

call WriteBinB; if ebx is set to 2 then WriteBinB will output contents of eax as 16 bit binary format(for WORD ebx is set to 1)

; Print msg2 string

mov edx, OFFSET msg2

call WriteString

; Now check individual bits of the register AX.To check status of individual bits in each iteration AND registers bxand cx.Since only first bit of cx is set so if the first bit of bx is also set then AND will result into 1 otherwise 0.

mov edx, 1; counter to iterate over 16 bits

mov bx, ax; copy the number in bx

mov cx, 0x0001; 0000 0000 0000 0001

CHECK:

cmp edx, 17

je END; If counter reaches 17 then end the loop

and cx, bx; AND of cxand bx

jz ZERO; If this results in 0 then bit is not set, jump to ZERO

jmp ONE; jump to ONE

ZERO :

; iii) If 0 then sprinkler is OFF, display sprinkler number

mov eax, edx

call WriteInt; Display the sprinkler number

shr bx; Right shift bx to check next bit

inc edx; increment the counter

jmp CHECK

ONE :

; If 1 then sprinkler is ON, increment the counter

mov eax, defSpr

inc eax

mov defSpr, eax

shr bx; Right shift bx to check next bit

inc edx; increment the counter

jmp CHECK

END :

; ii) Display how many sprinklers are ON

; Print msg3

mov edx, OFFSET msg3

call WriteString

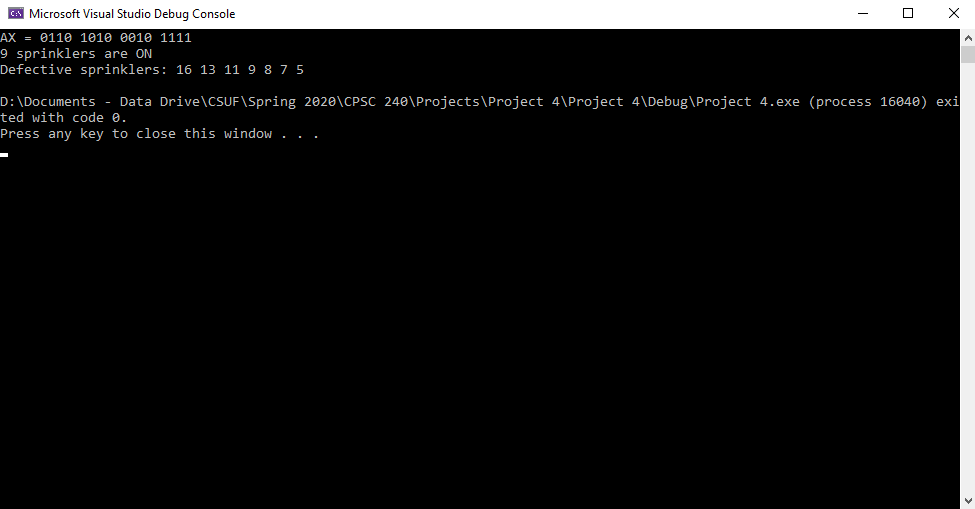
; print count

mov eax, defSpr

call WriteInt

\_exit

}



// Question 3

\_asm

{

; i) Print "Ax="

mov edx, OFFSET msg1

call WriteString

; Display register AX

mov ax, 0x6a2f

mov ebx, TYPE 2

call WriteBinB;

; Print msg2 string

mov edx, OFFSET msg2

call WriteString

mov edx, 1; counter to iterate over 16 bits

mov bx, ax; copy the number in bx

mov cx, 0x0001; 0000 0000 0000 0001

CHECK:

cmp edx, 17

je END; If counter reaches 17 then end the loop

and cx, bx; AND of cxand bx

jz ZERO; If this results in 0 then bit is not set, jump to ZERO

jmp ONE; jump to ONE

ZERO :

; iii) If 0 then sprinkler is OFF, display sprinkler number

mov eax, edx

call WriteInt; Display the sprinkler number

shr bx; Right shift bx to check next bit

inc edx; increment the counter

jmp CHECK

ONE :

; If 1 then sprinkler is ON, increment the counter

mov eax, defSpr

inc eax

mov defSpr, eax

shr bx; Right shift bx to check next bit

inc edx; increment the counter

jmp CHECK

END :

; ii) Display how many sprinklers are ON

; Print msg3

mov edx, OFFSET msg3

call WriteString

; print count

mov eax, defSpr

call WriteInt

\_exit

}

