Project 5

Students: Max Krochman, Tom Abrahams

Due 3/2/2017

Assembly 240

//

//Assembly 240 Project 5 Part 1

//Students Max Krochman, Tom Abrahams

//

#include "stdafx.h"

#include <iostream>

using namespace std;

short maxBits = 16, counter = 0, numOfPrinters = 0, numOfFloopy = 0;

int numOfMemory = 0;

void displayOutput()

{

cout << "Number of Printers: " << numOfPrinters

<< endl << "Number of Floppy: " << numOfFloopy

<< endl << "Number of Memory: " << numOfMemory << "K" << endl;

cin >> numOfPrinters;

}

int main()

{

\_\_asm {

Printer:

inc counter;//counter from 0 to 1

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 14;// shift ax right 14 positions

and ax, 1;// find value of 15th bit

imul ax, counter;//Converts vaule from base 2 to 10

add numOfPrinters, ax;// add the first part of the number of printers to the final value

inc counter;//counter from 1 to 2

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 15;// shift ax right 15 positions

and ax, 1;// find value of 16th bit

imul ax, counter;//Converts vaule from base 2 to 10

add numOfPrinters, ax;// add the second part of the number of printers to the final value

mov counter, 0;//Reset counter for later use

////////////////////////////////////////////////////////////////

Floopy:

inc counter;//counter from 0 to 1

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 6;// shift ax right 6 positions

and ax, 1;// find value of 6th bit

imul ax, counter;//Converts vaule from base 2 to 10

add numOfFloopy, ax;// add the first part of the number of floppies to final value

inc counter;//counter from 1 to 2

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 7;// shift ax right 7 positions

and ax, 1;// find value of 7th bit

imul ax, counter;//Converts vaule from base 2 to 10

add numOfFloopy, ax;// add the second part of the number of floopies to the final value

inc numOfFloopy;// floopy value is 1+ reg value, adding one to normalize

mov counter, 0;//Reset counter for later use

////////////////////////////////////////////////////////////////

Memory:

inc counter;

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 2;

and eax, 1;

imul ax, counter;//Converts vaule from base 2 to 10

add numOfMemory, eax;

inc counter;

mov ax, 1100111010011100b; // starts off the program with the proper value in the AX register

shr ax, 3;

and eax, 1;

imul ax, counter;//Converts vaule from base 2 to 10

add numOfMemory, eax;

cmp numOfMemory, 0;

je SixteenK;

cmp numOfMemory, 1;

je ThirtyTwoK;

cmp numOfMemory, 2;

je FourtyEightK;

cmp numOfMemory, 3;

je SixtyFourK;

jmp Exit;

////////////////////////////////////////////////////////////////////

SixteenK:

mov numOfMemory, 16;

jmp Exit;

////////////////////////////////////////////////////////////////////

ThirtyTwoK:

mov numOfMemory, 32;

jmp Exit;

////////////////////////////////////////////////////////////////////

FourtyEightK:

mov numOfMemory, 48;

jmp Exit;

////////////////////////////////////////////////////////////////////

SixtyFourK:

mov numOfMemory, 64;

jmp Exit;

////////////////////////////////////////////////////////////////////

Exit:

call displayOutput;

}

return 0;

}



// Assembly project 5 part 2.cpp : Defines the entry point for the console application.

//

//

//Assembly 240 Project 5 Part 2

//Students Max Krochman, Tom Abrahams

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int userPin = 0, pinSum = 0, counter = 0;

char errorAnswer = ' ';

void userInput()

{

cout << "Enter a four digit int PIN number: ";

cin >> userPin;

cout << endl;

}

void errorMsg()

{

cout << userPin << " is not a valid PIN number" << endl;

cout << "Do you want to try again(y/n)? ";

cin >> errorAnswer;

cout << endl;

}

void errorMsgToBig()

{

cout << userPin << " is greater than 4 digits" << endl;

cout << "Do you want to try again(y/n)? ";

cin >> errorAnswer;

cout << endl;

}

void finaOutput()

{

cout << userPin << " is a valid PIN number" << endl;

cin >> userPin;

}

int main()

{

\_\_asm {

Start:

mov counter, 0;

mov errorAnswer, ' ';

mov pinSum, 0;

call userInput;

mov eax, userPin;

cmp eax, 9999;

jle under\_10000;

over\_9999:

call errorMsgToBig;

cmp errorAnswer, 'y'

je Start;

cmp errorAnswer, 'Y'

je Start;

jmp ErrorEnd;

under\_10000:

mov errorAnswer, ' ';

inc counter;

cdq;

mov ebx, 10;

idiv ebx;

add pinSum, edx;

mov ecx, counter;

cmp counter, 4;

je testForOdd;

jmp under\_10000;

testForOdd:

mov eax, pinSum;

and eax, 1;

cmp eax, 1;

je oddNumb;

jmp Exit;

oddNumb:

call errorMsg;

cmp errorAnswer, 'y'

je Start;

cmp errorAnswer, 'Y'

je Start;

jmp ErrorEnd;

Exit:

call finaOutput;

ErrorEnd:

}

return 0;

}



// Assembly project 5 part 3.cpp : Defines the entry point for the console application.

//

//

//Assembly 240 Project 5 Part 3

//Students Max Krochman, Tom Abrahams

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int counter = 0, onSprinkelers =0;

void startOutput()

{

cout << "AX=";

}

void outputOne()

{

cout << "1";

}

void outputZero()

{

cout << "0";

}

void outputOnSprinkelers()

{

cout << endl << onSprinkelers << " sprinklers are ON" << endl;

}

void outputBroken()

{

cout << "Defective sprinklers: ";

}

void outputDefectiveNumbers()

{

cout << counter << " ";

}

int main()

{

\_\_asm {

mov eax, 0;

mov bx, 16;

call startOutput;

cloop:

mov ax, 0x6A2F;//store proper value in AX reg

inc counter;//Add one to counter

dec bx;//subtract 1 from BX value

bt ax, bx;//Test(with Bit Test) the bx'th bit in the AX register and stores it in the CF(carry flag)

jc cone;//True if CF(carry flag is value 1/False if 0)

zero://False section

call outputZero;//Print out '0'

jmp check;//jump to check for loop

cone://True section

call outputOne;//OUtput 1

inc onSprinkelers;//add one to num of sprinkelers working

check:

cmp counter, 16;

je on;

jmp cloop;

on:

call outputOnSprinkelers;

defective:

mov counter, 0;

call outputBroken;

loop1:

mov ax, 0x6A2F;//store proper value in AX reg

inc counter;//Add one to counter

dec bx;//subtract 1 from BX value

bt ax, bx;//Test(with Bit Test) the bx'th bit in the AX register and stores it in the CF(carry flag)

jc one1;//True if CF(carry flag is value 1/False if 0)

zero1:

jmp check1;

one1:

call outputDefectiveNumbers;

check1:

cmp counter, 16;

je Exit;

jmp loop1;

Exit:

}

cin >> counter;

return 0;

}



// Assembly project 5 part 4.cpp : Defines the entry point for the console application.

//

//

//Assembly 240 Project 5 Part 4

//Students Max Krochman, Tom Abrahams

//

#include "stdafx.h"

#include <iostream>

using namespace std;

short var = 0;

int counter = 0;

void elevatorMsg()

{

cout << endl << "Eleavator will stop at floors NO. ";

}

void outPutFloors()

{

cout << counter << " ";

}

void outputOne()

{

cout << '1';

}

void outputZero()

{

cout << "0";

}

void outputAX()

{

cout << "AX= ";

}

int main()

{

\_\_asm {

mov bx, 16;

call outputAX;

Cloop1:

mov ax, 1001000100001100b;

inc counter;//Add one to counter

dec bx;//subtract 1 from BX value

bt ax, bx;//Test(with Bit Test) the bx'th bit in the AX register and stores it in the CF(carry flag)

jc one1;//True if CF(carry flag is value 1/False if 0)

zero0:

call outputZero;

jmp check1;

one1:

call outputOne;

jmp check1;

check1:

cmp counter, 15;

jle Cloop1;

mov bx, 0;

mov counter, 1;

call elevatorMsg;

Cloop:

mov ax, 1001000100001100b;

inc counter;//Add one to counter

inc bx;//subtract 1 from BX value

bt ax, bx;//Test(with Bit Test) the bx'th bit in the AX register and stores it in the CF(carry flag)

jc yesStop;//True if CF(carry flag is value 1/False if 0)

noStop:

jmp check;

yesStop:

call outPutFloors;

jmp check;

check:

cmp counter, 16;

jle Cloop;

}

cin >> var;

return 0;

}

