|  |  |
| --- | --- |
| **Topic** | Practical Assignment 3 Cover Sheet |
| **Assignment Type** | 🗷 Assessed 🞏 Non-assessed  🗷 Individual 🞏 Group |
| **Module** | CSE101 Computer Systems |
| **Due Date** | November 7th, 2018 (Wednesday) |
| **Student ID** | 1717576 |
| **Student Name** | Minhao Jin |
| **Submission Date** | November 7th, 2018 (Wednesday) |

**Declaration on Plagiarism and Collusion**

I have read and understood the definitions of plagiarism and collusions as described in the University’s Code of Practice on Assessment. As such, I certify the work presented in this report/assignment has been written solely by me and in my own words (except where references and acknowledgments are clearly defined). I agree to accept disciplinary actions should I be caught with the serious offence of plagiarism and/or collusion.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **For Academic Use** | **Date Received** | **No. of Days Late** | **Penalty** |
|  |  |  |

**Program Listing**

*<*

#include "stdafx.h"

#include <stdlib.h>

int main()

{

int loopCounter=0;

char character;

int numberCounter = 1;

int entries = 0;

int UCcount = 0;

int LCcount = 0;

int EVENcount = 0;

int ODDcount = 0;

char df[] = "%d";

char input\_cf[] = "\r%c";

char newline[] = "\n";

char msg1[] = "Enter number of loops: ";

char enter\_format[] = "Enter alphanumeric character %d:";

char wrongnum[] = "Please enter a positive integer.\n";

char end[] = "Program ends.\n";

char entriesresult[] = "Number of entries is %d \n";

char uppercaseresult[] = "Number of uppercase numbers is %d \n";

char lowercaseresult[] = "Number of lowercase numbers is %d \n";

char evenresult[] = "Number of even numbers is %d \n";

char oddresult[] = "Number of odd numbers is %d \n";

char even[] = "Even\n";

char odd[] = "Odd\n";

char uppercase[] = "Uppercase\n";

char lowercase[] = "Lowercase\n";

\_asm {

looptimes: //Function: to ensure the input is a positive integer

lea eax, msg1;

push eax;

call printf;

add esp, 4; //Print "Enter number of loops:" on the screen

lea eax, loopCounter;

push eax;

lea eax, df;

push eax;

call scanf\_s;

add esp, 8; //Input the times for loop

mov eax, loopCounter;

cmp eax, 0; //Judge whether the input number is positive or negative

jle wrongnumber; //Jump to "wrongnumber" if the input number is negative

jg start; //Jump to "start" if the input number is positive

wrongnumber:

lea eax, wrongnum;

push eax;

call printf;

add esp, 4; //Print "Please enter a positive integer." on the screen

mov eax, loopCounter;

cmp eax, 0;

jle looptimes; //If the input number is negative, go back to the "looptimes" for another input

start: //Function: Save the value of "loopCounter" in ecx

mov ecx, loopCounter;

judgeloop: //Loop starts here!

push ecx;

mov eax, numberCounter;

push eax;

lea eax, enter\_format;

push eax;

call printf;

add esp, 8; //Print "Enter alphanumeric character 'numberCounter':" on the screen. "numberCounter" is the current loop times.

inc numberCounter; //Add 1 to the variable "numberCounter"

push 1;

lea eax, character;

push eax;

lea eax, input\_cf;

push eax;

call scanf\_s;

add esp, 12; //Input one character

movzx eax, byte ptr[character]; //Save the value of "character" in eax

cmp eax, '\*'; //Compare eax with '\*'

je finish; //if the input character equals '\*', jump to the fuction named "finish"

cmp eax, '9'; //Compare eax with '9'

jle numberjudge; //if the input character is less than or equal to the value of '9', that means it is a number. Then jump to the fuction named "numberjudge"

cmp eax, 'Z'; //Compare eax with 'Z'

jle labeluppercase; //if the input character is less than or equal to the value of 'Z', that means it is an uppercase letter. Then jump to the fuction named "labeluppercase"

cmp eax, 'a'; //Compare eax with 'a'

jge labellowercase; //if the input character is greater than or equal to the value of 'a', that means it is an lowercase letter. Then jump to the fuction named "labellowercase"

numberjudge: //Function: to judge whether the number which the character represents is even or odd

mov ebx, 2; //Let ebx=2

div ebx; //eax/ebx

test edx, edx; //Judge the remainder of eax/ebx

jz labelEven; //if the remainder is 0, jump to the function named "labelEven"

jne labelOdd; //if the remainder is not 0 (remainder = 1), jump to the function named "labelOdd"

output: //Function: add a newline between two character tests

lea eax, newline;

push eax;

call printf;

add esp, 4;

pop ecx;

loop judgeloop; //Loop ends here

jmp finish; //If the loop ends, jump to the function named "finish" immediately

labelEven: //Function: if the number which the character represents is even, do following things

lea eax, even;

push eax;

call printf;

add esp, 4; //Print "Even" on the screen to tell the user that the number which the character represents is even

inc EVENcount; //Add 1 to the variable "EVENcount"

inc entries; //Add 1 to the variable "entries"

jmp output; //Jump to the function named "output" and go back to the loop

labelOdd: //Function: if the number which the character represents is odd, do following things

lea eax, odd;

push eax;

call printf;

add esp, 4; //Print "Odd" on the screen to tell the user that the number which the character represents is odd

inc ODDcount; //Add 1 to the variable "ODDcount"

inc entries; //Add 1 to the variable "entries"

jmp output; //Jump to the function named "output" and go back to the loop

labeluppercase: //Function: if the letter which the character represents is uppercase, do following things

lea eax, uppercase;

push eax;

call printf;

add esp, 4; //Print "Uppercase" on the screen to tell the user that the letter which the character represents is uppercase

inc UCcount; //Add 1 to the variable "UCcount"

inc entries; //Add 1 to the variable "entries"

jmp output; //Jump to the function named "output" and go back to the loop

labellowercase: //Function: if the letter which the character represents is lowercase, do following things

lea eax, lowercase;

push eax;

call printf;

add esp, 4; //Print "Lowercase" on the screen to tell the user that the letter which the character represents is lowercase

inc LCcount; //Add 1 to the variable "LCcount"

inc entries; //Add 1 to the variable "entries"

jmp output; //Jump to the function named "output" and go back to the loop

finish: //Function: Print out all the test results

mov eax, entries;

push eax;

lea eax, entriesresult;

push eax;

call printf;

add esp, 8; //Print out the value of "entries"

mov eax, UCcount;

push eax;

lea eax, uppercaseresult;

push eax;

call printf;

add esp, 8; //Print out the value of "UCcount"

mov eax, LCcount;

push eax;

lea eax, lowercaseresult;

push eax;

call printf;

add esp, 8; //Print out the value of "LCcount"

mov eax, EVENcount;

push eax;

lea eax, evenresult;

push eax;

call printf;

add esp, 8; //Print out the value of "EVENcount"

mov eax, ODDcount;

push eax;

lea eax, oddresult;

push eax;

call printf;

add esp, 8; //Print out the value of "ODDcount"

lea eax, end;

push eax;

call printf;

add esp, 4; //Print "Program ends" on the screen

}

system("Pause");

return 0;

}

*>*