PDS LAB [Date: 27th Aug 2024]

Assignment – 3 [Loops & Functions]

Instructions:

1. Create a directory named as Lab-3.

2. Give the name of the program as <p>.c where <p> implies the problem number, like 1.c, 2.c, 3.c, etc. Store all the programs of this week under this directory.

3. You should upload all .c files (1.c, 2.c, 3.c ….) to the Moodle course web page latest by 5.00 PM (without penalty). The cutoff time will be till 5.15 PM with a penalty of 25% on your secured marks (i.e., if you secured 80 marks, after penalty you will get 60 marks). Beyond 5.15 PM, the moodle system will not allow you to submit, as a result you will get zero.

---

p1] Write a C program to print the following pattern of numbers by giving the number of rows N as input in the range 1 <= N <= 30. User is supposed to enter the number of rows as positive odd integer within the specified range. In case of wrong entry, the program should prompt the user about the error and suggest him/her to enter the correct input. The program should not exit in case of invalid input. **[30 Marks]**

Example-1: N = 5

```
    1
  2 3 2
3 4 5 4 3
  2 3 2
    1
```

Example-2: N = 7

```
      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
  3 4 5 4 3
    2 3 2
      1
```

---

p2] In number theory, a perfect number is a positive integer, where sum of its positive divisors (excluding the number itself) is equal to the number itself. For example, 28 is a perfect number where sum of its 1+2+4+7+14 = 28. **[30 Marks]**

Write a program in C consisting two user defined functions to print all perfect numbers in a given range provided by the user through keyboard. Through main(), the user-defined functions will be called for performing the task. The user-defined functions are given below:

a) Write a function ***checkPerfect(int x)*** that returns 1 if x is a perfect number else returns 0.

b) Write a function ***ListPerfectNumbers(int, int)*** that prints all the perfect numbers in the range provided by the user.

The program should alert the user in case if the user provides an invalid input. The program should not exit in case of invalid input.

Example :

Input the range : 100 10000
The perfect numbers between 100 and 10000 are : 496 8128

Input the range : 100 -10000
Input Range is incorrect. Enter the correct range
The perfect numbers between 100 and 10000 are : 496 8128

---

p3] Write a C program to find the (i) number of characters, (ii) number of words, (iii) size of each word, (iv) max and minimum size of the entered words and (v) number of pairs of successive words of equal size. Here, the user is supposed to enter only alphabets and space (for word separation) through keyboard and the input to be terminated with enter key. While entering the input, if the user enters any non-alphabet character (other than enter and space keys), the program should print the words till the correct entry and prompt the user about the error and instruct the user to enter the correct input again. You should use appropriate prompts for better user interface.

In this program, use the following three user defined functions (i) ***int min_max(int length, int max, int min)*** [min_max function is used to return +1 if the length of the current word is greater than or equal to max and return -1 if the length of the current word is less than or equal to min, otherwise it returns zero] (ii) ***int word_equal(int pl, int cl)*** [This function returns 1, if previous and current words are of same length, else it returns 0] and (ii) ***void output(int nc, int nw, int max, int min, int ne)*** [This function display the output in 5 lines: number of characters (without spaces), number of words, maximum length of the word, minimum length of the word and number of pairs of successive words of equal size]. You need to define the functions and call these functions from main() function appropriately to run the program and display the results as per the requirement. **[40 Marks]**

Example-1:

Enter the input text using only alphabets :
asd asd wwdee xd

word[1] = asd
Length of word[1] = 3

word[2] = asd

Length of word[2] = 3

word[3] = wwdee
Length of word[3] = 5

word[4] = xd
Length of word[4] = 2

Number of characters = 13
Number of words = 4
Maximum size of the words = 5
Minimum size of the words = 2
Number of pairs of successive words of equal size = 1

Example-2:

Enter the input text using only alphabets :
asd asd wwdee xd xd 1fg

word[1] = asd
Length of word[1] = 3

word[2] = asd
Length of word[2] = 3

word[3] = wwdee
Length of word[3] = 5

word[4] = xd
Length of word[4] = 2

word[5] = xd
Length of word[5] = 2

word[6] =
Enter the correct input again

THE END