PDS Lab, Section - 17, Date: 11th Jan 2020
Assignment - 5 [Functions]

Instructions

----

1. **Create a directory named as <rollno>_A5, where <rollno> is your roll number.**
2. **Give the name of the program as <p>.c where <p> implies the problems number, like 1.c 2.c 3.c etc. Store all the program under this Assignment in the directory <rollno>_A5**
3. **Zip the entire directory <rollno>_A5.**
4. **You should upload your zipped file <rollno>_A5.zip to the Moodle course web page latest by 5 PM (without penalty). The cutoff time will be till 5.30 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 5.30 PM, the moodle system will not allow you to submit, as a result you will get zero.**
5. **Each question carries 25 marks.**

Q1. Write a program in C to print all perfect numbers in a range given by the user. In this program you need to use two C functions. The first function will check whether the given number is perfect or not, and return 1 (in case of perfect number) or 0 (otherwise). The second function should print all the perfect numbers in the range provided by the user, by using the first function. The user has to specify the range of numbers using two numbers (through keyboard), say p and q, where p and q are lower and upper bounds of the range. Through the main function, the user has to call the second function by passing the range as arguments to print all perfect numbers within the range.

Note: *A number is said to be perfect, if the sum of all the divisors of a number equal to the number itself.*

In this problem, consider the numbers as positive integers.

Eg (i) 6 = 1,2,3 (1+2+3 = 6); (ii) 28 = 1,2,4,7,14 (1+2+4+7+14= 28)

Q2. Write a C program that accepts the input string (alphanumeric characters), and display the output string by removing all the occurrences of a given character. For carrying out this task, you need to write two C functions. The first C function is used for inputting the string with recursive calls. The string entry will be controlled/terminated by enter(\n) key. The second function will take an array which contains the input string, the specific character and an empty string as input. The function should remove all the occurrences of a specific character in the input string, and the remaining string is stored in the other array. From the main(), the user has to call these two functions and then both input and output (after removing the specific character) strings to be printed in the main().

EX-1:

Enter the input string: apple

Specific Character: p

Input : apple

Output : ale

EX-2:

Enter the input string: apple

Specific Character: x

Input : apple

Output : apple

Q3. Write a C program to generate the strings with alphanumeric characters using the following constraints:

(i) Consider an integer array loaded by the user from main().

(ii) Input two alphanumeric characters through the keyboard.

(iii) Write a C function to generate the strings with the following features:

(a) Number of strings equal to number of elements of an input array.

(b) The lengths of the strings are restricted by the values of the array elements.

(c) The characters within the string should have the following properties: If the length of the string is even, the first half of the string is repeated with 1st character and 2nd half of the string is repeated 2nd character. Otherwise, the 1st character has to repeat one time more than the 2nd character.

You need to call this C function from the main() and pass the desired four parameters.

Example:

Input array length N = 4

Input Array Elements = p[4] = {2,9,3,10}.

Enter two characters = 'a', 'b'

Output: ab, aaaaabbbb, aab, aaaaabbbbb

Q4. Write a C program with the functions mentioned below to perform the following:

(i) Take two positive integers (i1,i2) less than 50,000 as an input, and find their binary representations and store it in arrays (b1[], b2[]) using integer to binary conversion function void int_to_bin(int, int[]). For converting each integer you need to call the function separately.

(ii) Perform XOR operation (x[] = b1[] xor b2[]) using the function void xor(int [], int [], int[]).

(iii) Convert x[] into integer using the function int bin_to_int(int []).

Call these functions through main() to perform the sequence of above mentioned operations.

Ex-1:

Input: 5 5

b1 = b2 = 0000000000000101

X = 0000000000000000

Integer value of x = 0.

Ex-2:

Input:

Input : 456, 321

b1 = 0000000111001000

b2 = 0000000101000001

X = 0000000010001001

Integer value of x = 137