In The Name Of God

( Problem set - 6 )

1 - write a function that gets multiple lines of input from the user and then converts the “newline” , “tab” , “space” into visible escape sequences such as “\n” , “\t” , “b”.you have to use switch case . write a function for other direction as well ( for example converts “\t” to tab ).

2 - add commands to print the top elements of the stack without popping it , swap the top two elements , and clearing the whole stack .

1. print\_top()
2. swap\_two\_top()
3. clear\_stack()

3 - write a simple calculator . the calculator should be able to apply precedence and read parentheses and make decision that which operation will be done first .

>>>(2\*4) + ( 9 \* 8 ) + 2-9\*8

>>>Output : 10

4 - take a look at math.h library and see the available functions.

5 - write a function that takes a number n as input and returns the smallest number that can be made with the digits of n .[ you have to use **only** char arrays ]

Ex :

>>input : 82340325

>> output :20233458

Ex2 :

>> input : 1234

>> output : 1234

6 - تابعی بنویسید که بزرگترین عامل اول مشترک دو عدد را پیدا کند

7 - write a program that evaluates the following expression

display the result in integer format.

ans = 7 times 9 plus ( 19 divided by 5 ) modulo 2

int ans = 7 \* 9 + ( 19 / 5 ) % 2

( operators are + - \* / ( ) %)

( phrases are : times , plus , minus , divided by , modulo , ( , ) )

8 - write a function that gets the output of question 5 and calculates the result .

**float calculator ( char sequence[] ) ;**

// char seq[] = “ 7 \* 9 + ( 19 / 5 ) % 2 “

printf(“%f ” , Calculator ( seq ) );

//output : 64

9 - write a simple calculator that can calculate the given sequence by precedence of operators . the operators can be :

\* , / , + , - , ( , ) , ^

^ = power function => 2^3 = 8

10 - write a function that would take a set and writes it’s subsets.

**( Recursive )**

Ex : set1 = { 1 , 2 , 3 }

Subset = { null , {1} , {2} , {3} , {1,2} , {1,3} , {2,3} ,{1,2,3} }

11 - write the **fibonacci** and **factorial** functions **recursively** .

**( Recursive )**

12 - Get scientific number

Using only the getchar () function for getting Input, then Write a function that gets a number in scientific notation and returns it as a double and print it using putchar function.

Prototype: double next\_sc (char num[]); Examples:

next\_sc (“12e-1”) --> next\_sc (“12e9”) -->

next\_sc (“0e+999”) -->

returns: 1.2

returns: 12000000000 returns: 0

13 - Write a program to ask user 2 big numbers **a** and **b** (in string format) using getchar() function and compute **result = a + b** and print it using putchar function.

**Note** : the numbers can be **negative** too.

Sample output:

Enter the first number : 555555555

Enter the second number : 999999999

Sum is : 1555555554

14 - in order to face more complexity , write **BigMul** function .

15 - تابعی بنویسید که از کاربر یه معادله درجه دو را گرفته و آنرا حل کند.

ورودی نمونه : x^2 + 4x + 4

خروجی : +2 و -2

16 - Two words are anagram if they have same letters in same or different orders, write a function to determine whether two strings are anagram or not, then return 0 if they’re not and 1 otherwise. Declaration:

int isAnagram (char string1[], char string2[]) ;

Ex :

>> “ nima “ and “mina “ are anagram .

17 - assume we have an array of integers , write a function that finds the subarray with maximum summation of elements .

**( Recursive ) & ( Non-Recursive )**

>> Ex : {13 , -3 , -25 , 20 , -3 , -16 , -23 , 18 , 20 , -7 , 12 , -5 , -22 , 15 , -4 , 7}

>> from index 7 through index 10 , we have the subarray with maximum summation which is 43 .

The output of this program should be 43

If you want to face more complexity , the output shall show :  
>> from index **7** through index **10** we have max : **43**