DAY 4

Random number generator

#include<iostream>

#include<ctime>

#include<stdlib.h>

using **namespace** std;

**int** main()

{   system("cls");

**int** max,random,i;

for (i = 0; i <= 4; i++)

{

    srand(time(0));

    random = (rand() % (i+4));

    system("Color 06");

    cout<<"\n\nrandom number : "<<random;

}

    return 0;

}

--------------------------------------------------------------------------------------------------------------------------------------

#DSA 1 : add numbers from 2 lists

class Solution:

def addTwoNumbers(self, l1, l2):

"""

:type l1: ListNode

:type l2: ListNode

:rtype: ListNode

"""

result = ListNode(0)

result\_tail = result

carry = 0

while l1 or l2 or carry:

val1 = (l1.val if l1 else 0)

val2 = (l2.val if l2 else 0)

carry, out = divmod(val1+val2 + carry, 10)

result\_tail.next = ListNode(out)

result\_tail = result\_tail.next

l1 = (l1.next if l1 else None)

l2 = (l2.next if l2 else None)

return result.next

--------------------------------------------------------------------------------------------------------------------------------------

DSA 2 : integer to romans

class Solution:

def intToRoman(self, num: int) -> str:

val = [

1000, 900, 500, 400,

100, 90, 50, 40,

10, 9, 5, 4,

1

]

syb = [

"M", "CM", "D", "CD",

"C", "XC", "L", "XL",

"X", "IX", "V", "IV",

"I"

]

roman\_num = ''

i = 0

while num > 0:

for \_ in range(num // val[i]):

roman\_num += syb[i]

num -= val[i]

i += 1

return roman\_num

--------------------------------------------------------------------------------------------------------------------------------------

DSA 3 # letter combination of phone numbers – c++

class Solution {

public:

vector<string> abc {"abc","def","ghi","jkl","mno","pqrs","tuv","wxyz"};

void dp(int i,string dig,string curr,vector<string>& ans)

{

if(i==dig.size())

ans.push\_back(curr);

else

{

int j=dig[i]-'0';

//going through all possible orientations for the specific digit

//j-2 as each numbers corresponding possible digits are stored at numbers-2 index

for(int a=0;a<abc[j-2].size();a++)

dp(i+1,dig,curr+abc[j-2][a],ans);

}

}

vector<string> letterCombinations(string digits) {

vector<string> ans;

if(digits.size()==0) return ans;

dp(0,digits,"",ans);

return ans;

}

};

--------------------------------------------------------------------------------------------------------------------------------------

DSA 4 : generate parentheses - Python

class Solution:

def generateParenthesis(self, n: int) -> List[str]:

res = []

def helper(s, m, d):

if len(s) == 2\*n:

res.append(s)

return

if d > m:

return

if m == d:

helper(s+'(', m+1, d)

elif m == n:

helper(s+')', m, d+1)

else:

helper(s+'(', m+1, d)

helper(s+')', m, d+1)

helper('', 0, 0)

return res

--------------------------------------------------------------------------------------------------------------------------------------

DSA 5 : Longest palindromic substring ( code in python 3)

class Solution:

def longestPalindrome(self, s: str) -> str:

for x in reversed(range(1,len(s)+1)):

for y in range((len(s)+1)-x):

g=s[y:x+y]

if g[0]!=g[-1]:pass

elif g==g[::-1]:return g