Recursion (class 2)

11-03-2023

Date.....

Ques) Climbing stairs.

You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps.

In how many ways can you climb to the top.

for example:-

Input, n = 2

output, 2

Explanation, There are two ways to climb to the top.

2 ways to climb to stair 2
- way 1, 0 to 1 + 1 to 2
  1 step + 1 step,
  i.e., 2 steps.
- way 2, 2 steps.

for example:-

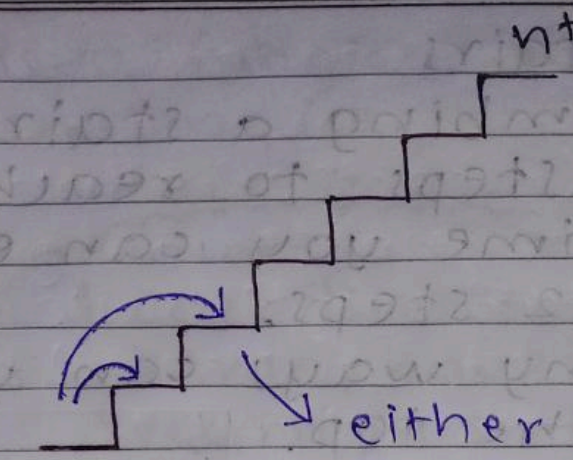
Input, n = 3

output, 3

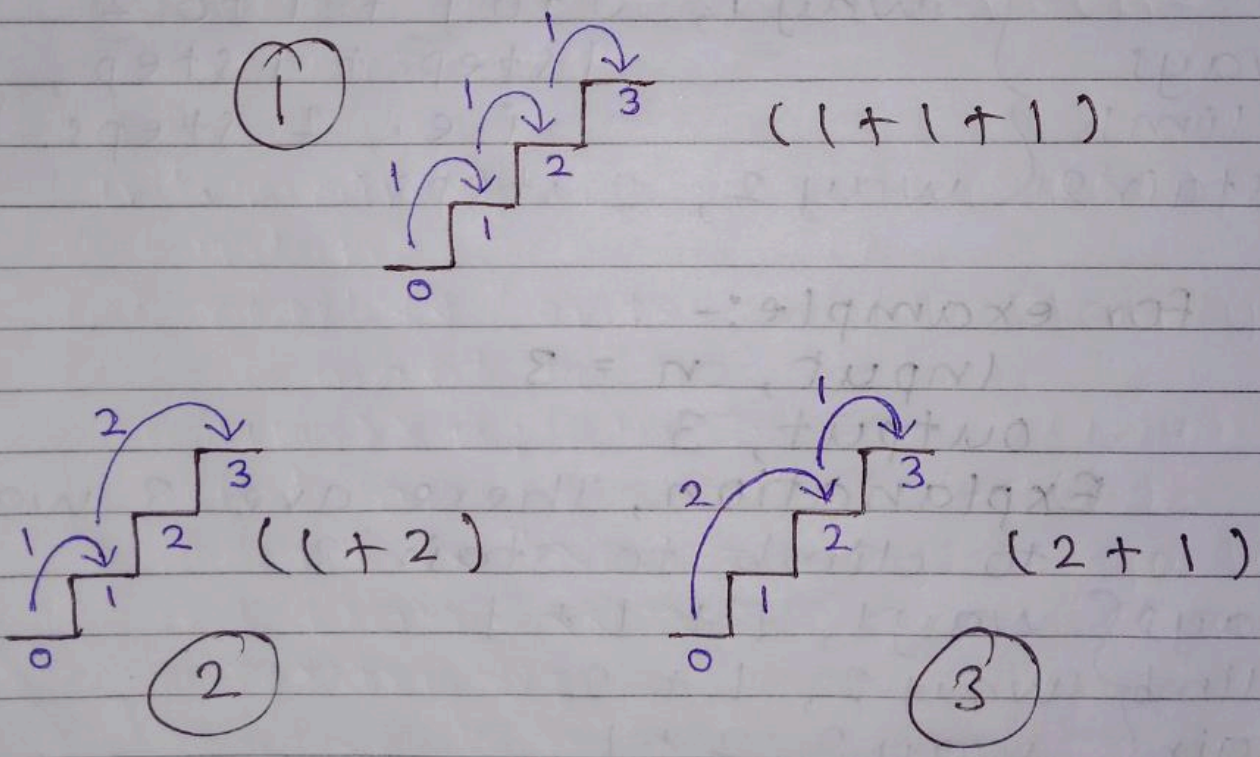
Explanation, There are 3 way to climb to stair 3.

3 ways to climb to stair 3.
- way 1, 1 + 1 + 1
- way 2, 1 + 2
- way 3, 2 + 1

Date.....



To climb at stair 3, the no. of ways are :-



In this problem, rather than we think that how to go from 0th stair to nth stair. we directly think that how we get on nth stair.

Date.....

n^{th} stair pe pahuchne ke 2 ways hai, either we came from $n-1^{th}$ stair by taking 1 step or we came from $n-2^{th}$ stair by taking 2 steps. There is no other way to reach at n^{th} stair.

$$climb(n) = climb(n-1) + climb(n-2)$$

↳ recursive relation for this question.

This recursive relation is same as of the Fibonacci series recursive relation.

Here, we just solved for only 1 case, i.e., for last n^{th} stair & rest for all stairs recursion automatically solve the problem.

Date.....

code :-

```
#include<iostream>
using namespace std;
int climbStairs(int n){
    if(n == 0){
        return 1;
    }
    if(n == 1){
        return 1;
    }
    int ans = climbstairs(n-1) +
              climbstairs(n-2);
    return ans;
}
int main(){
    int n;
    cout << "enter which stair you
        want to go: ";
    cin >> n;
    int ans = climbstairs(n);
    cout << "No. of ways to go to
        stair " << n << " are: " << ans;
    return 0;
}
```

1 way to go on 0th stair from 0th stair (jump)

1 way to go on 1st stair from 0th stair.

Date.....

Ques Traverse on array using recursion.

```
#include<iostream>
using namespace std;
//recursive approach
void printArray(int arr[], int n,
                int i)
{
    //base case
    if(i >= n){
        return;
    }
    cout << arr[i] << " ";
    printArray(arr, n, i+1);
}
int main()
{
    int arr[] = {10,20,30,40,50,60};
    int n = 6;
    int i = 0;
    printArray(arr, n, i);
    return 0;
}
```

Date.....

Ques. find maximum number in an array using recursion.

```
#include<iostream>
using namespace std;
void findMax(int arr[], int n,
             int &maxi, int i)
{
    //base case
    if(i >= n){
        return;
    }
    if(arr[i] > maxi){
        maxi = arr[i];
    }
    findMax(arr, n, maxi, i+1);
}
int main()
{
    int arr[] = {10, 20, 70, 68, 80, 49, 15};
    int n = 7;
    int i = 0;
    int maxi = INT_MIN;
    findMax(arr, n, maxi, i);
    cout << "Max no. is " << maxi;
    return 0;
}
```

passed as reference for change in the original value.

Date.....

Ques. Find minimum number in an array using recursion.

```
#include<iostream>
using namespace std;
void findMin(int arr[], int n,
             int &mini, int i)
{
    // base case
    if(i >= n){
        return;
    }
    if(arr[i] < mini){
        mini = arr[i];
    }
    findMin(arr, n, mini, i++);
}
int main()
{
    int arr[] = {10, 20, 70, 3, 2, 80};
    int n = 6;
    int i = 0;
    int mini = INT_MAX;
    findMin(arr, n, mini, i);
    cout << "Min no. is " << mini;
    return 0;
}
```

Date.....

Ques Traverse string using Recursion.

```
#include<iostream>
using namespace std;
void findchar(string str, int n,
    char c, int i, int &charIndex)
{
    //base case
    if(i >= n){
        return;
    }
    if(str[i] == c){
        charIndex = i;
    }
    findchar(str, n, c, i+1, charIndex);
}
int main()
{
    string str = "baabubhaiya";
    int n = str.length();
    int i = 0;
    char c = 'y';
    int charIndex;
    findchar(str, n, c, i, charIndex);
    cout << "\nIndex where the char
      -acter " << c << " is found is : " <<
          charIndex << endl;
    return 0;
}
```

By reference

Teacher's Sign

Date.....

→ Print every index where the characters found.

```
if(str[i] == c){
    cout << "found at" << i;
}
```

Date.....

Ques Given a number 'n', print all digits of a number using recursion.

```
#include<iostream>
using namespace std;
void printDigits(int n)
{
    //base case
    if(n==0){
        return;
    }
    int newvalue = n/10;
    printDigits(newvalue);
    int digit = n%10;
    cout << digit;
    cout << endl;
}
int main()
{
    int n = 647;
    printDigits(n);
    return 0;
}
```

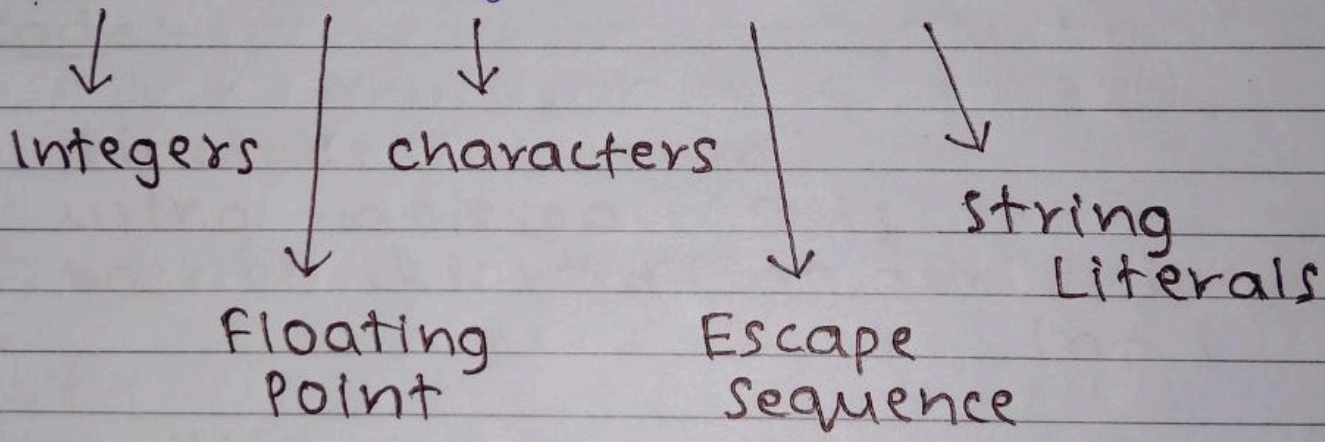
Teacher's Sign

In previous question, if given value of n is 0647. Then the digits printing are 4 2 3.

why?

Because, the first digit cannot be zero.
Integer literals begins with the digit 0 are interpreted as an octal integer literal rather than as a decimal integer literal.

There are many literals in C++ :-



(1) Integer Literal = An integer is a numeric literal (associate with numbers) without any fractional or exponential part. There are 3 types of integer literals in C++ :-
   i) decimal (base 10)
   ii) octal (base 8)
   iii) hexadecimal (base 16)

For example:-

decimal: 0, -9, 22, etc.
octal: 021, 077, 033, etc.
hexadecimal: 0x7f, 0x2a, etc.

In c++, octal starts with 0.

Hexadecimal starts with 0x.