GECG10069 (561085) F25: Introduction to Programming (C++)

Lab 7 : Functions

What you will learn from Lab 7

In this lab, you will learn different ways to pass and modify variables in functions, including call by value, using global variables, and call by reference.

TASK 7-1 : CALL BY VALUE V.S. GLOBAL VARIABLE V.S. CALL BY REFERENCE

✓ **Call By Value**: The values of the variables (x, y) in *main* are copied into the local variables (x, y) in the *swap* function.

➢ Any modification of the local variables (x, y) inside *swap* does **not** affect the original variables in *main*.

```
#include <iostream>
using namespace std;

void swap(int, int);
int main()
{
    int x = 5, y = 6;
    cout << "x = " << x << " and y = " << y << endl;
    swap(x, y);
    cout << "x = " << x << " and y = " << y << endl;
    return 0;
}
void swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

✓ **Global Variable**: Variables are defined outside of all functions and can be accessed directly by any function.

➢ Modifications inside a function will **change the global variables permanently**, affecting all functions that use them.

```
#include <iostream>
using namespace std;

void swap();
int x = 5, y = 6;
int main()
{
```

```
    cout << "x = " << x << " and y = " << y << endl;
    swap();
    cout << "x = " << x << " and y = " << y << endl;
    return 0;
}
void swap()
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

✓ **Call By Reference**: The function receives the references (aliases) of the variables in *main*.

- Any modification inside the *swap* function **directly affects** the original variables in *main*.

```
#include <iostream>
using namespace std;

void swap(int&, int&);
int main()
{
    int x = 5, y = 6;
    cout << "x = " << x << " and y = " << y << endl;
    swap(x, y);
    cout << "x = " << x << " and y = " << y << endl;
    return 0;
}
void swap(int& x, int & y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

Exercise 7-1 : Palindrome Number

Description -

Write a program to determine whether an integer is a palindrome number.

A palindrome number is defined as a number that reads the same forward and backward.

For example: 121, 12321, and 1221 are palindromes; 123 and 120 are not.

Input :

- A positive integer n ($1 \leq n \leq 1{,}000{,}000$).

Output :

- Print "Yes" if n is a palindrome.
- Print "No" otherwise.

Constraints :

- *You must finish the functions:*

```
int reverse(int n);

bool isPalindrome(int n);
```

Sample Testcases -

Sample Input - 1
12321
Sample Output - 1
Yes

Sample Input - 2
1234
Sample Output - 2
No

Exercise 7-2 : Climbing Stairs

Description -

You are climbing a staircase with n steps.

At each move, you can climb either 1 step or 2 steps.

However, you are not allowed to land on any step that is a multiple of 3.

Your task is to print all valid paths to reach exactly step n.

Each path should be represented as a sequence of steps you land on, starting from 0 (ground) and ending at n.

Input :

- One integer n ($1 \leq n \leq 10$).

Output :

- Print all valid paths, one per line.
- If no valid path exists, print " **no solution**".

Constraints :

- *You must solve this problem using recursion.*
- *Iterative solutions with loops controlling the path generation are not allowed.*
- *You may implement a helper function such as:*

```
void climb(int curr, int target, string path);
```

where path records the current sequence of steps.

Sample Testcases -

Sample Input - 1
4
Sample Output - 1
0 1 3 4 0 2 4

Sample Input - 2
5
Sample Output - 2 (Print NOTHING)
0 1 2 4 5 0 2 4 5

Sample Input - 3
6
Sample Output - 3
no solution