Assignment 1

Name: Tanish Majumdar Roll No.: 002311001077

2. Write a program that accepts two integers from keyboard, adds them and prints their values. Use `cin` and `cout`.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;
    cout << "Sum: " << a + b << endl;
    return 0;
}
```

3. Create a factorial table using `cout` as follows:

- 1! = 1
- 2! = 2
- 3! = 6
- 4! = 24
- 5! = 120
- 6! = 720

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

int main()
{
    int x = 1;
```

```
    for (int i = 1; i <= 6; i++)
    {
        x *= i;
        cout << i << "! = " << x << endl;
    }
    return 0;
}
```

4. Write a program to print 1 to 10 using a for loop. Declare the loop variable inside the for loop. Check the scope of this variable.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

int main()
{
    for (int i = 1; i <= 10; i++)
    {
        cout << i << " ";
    }
    // cout << i;  i is not defined outside the loop so gives error
    return 0;
}
```

5. Write a program to display Celsius to Fahrenheit conversion table using a `for loop`. Consider only 0° to 100° Celsius. Declare variables when they are used for the first time.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

int main()
{
    for (int i = 0; i <= 100; i++)
    {
```

```
        cout << "Celsius: " << i << " Fahrenheit: " << (i * 9 / 5) + 32 <<
endl;
    }
    return 0;
}
```

6. Write a program that defines a constant π and takes radius of a circle from keyboard and prints area of that circle.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
#define PI 3.14159265358979323846
using namespace std;

int main()
{
    int r;
    cin >> r;
    cout << "Area of circle with radius " << r << " is " << PI * r * r <<
endl;
    return 0;
}
```

7. Write a function that takes an integer and returns the factorial of that number. Declare function parameter as `const`. Call the function with some argument from main function, store the result and print it.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

ll factorial(const ll &n)
{
    if (n == 0)
        return 1;
    return n * factorial(n - 1);
}
```

```cpp
int main()
{
    ll n;
    cout << "Enter the number: ";
    cin >> n;
    cout << "Factorial of " << n << " is " << factorial(n) << endl;

    return 0;
}
```

8. Write a function `swap()` that takes two integer arguments and interchanges the values of those arguments using reference. Now in the main function, instantiate two integer variables with some values. Print their values. Call the swap function with these variables. Finally print the values of those variables. Check the result.

```cpp
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

void swapCustom(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}

int main()
{
    int a, b;
    cout << "Enter two numbers: ";
    cin >> a >> b;
    cout << "Before swapping: a = " << a << ", b = " << b << endl;
    swapCustom(a, b);
    cout << "After swapping: a = " << a << ", b = " << b << endl;

    return 0;
}
```

9. Now write another function swap() that takes two strings (character array) and interchanges them without reference parameters. Test this function using some arguments. Rewrite the function using reference parameters. Again test this function with some arguments.

```
#include <bits/stdc++.h>
using namespace std;

void swapString(char a[], char b[])
{
    char temp[100];

    for (int i = 0; a[i] != '\0'; i++)
    {
        temp[i] = a[i];
    }
    temp[strlen(a)] = '\0';

    for (int i = 0; b[i] != '\0'; i++)
    {
        a[i] = b[i];
    }
    a[strlen(b)] = '\0';

    for (int i = 0; temp[i] != '\0'; i++)
    {
        b[i] = temp[i];
    }
    b[strlen(temp)] = '\0';
}

void swapString(char* &a, char* &b)
{
    char temp[100];

    for (int i = 0; a[i] != '\0'; i++)
    {
        temp[i] = a[i];
    }
    temp[strlen(a)] = '\0';

    for (int i = 0; b[i] != '\0'; i++)
    {
        a[i] = b[i];
```

```
    }
    a[strlen(b)] = '\0';

    for (int i = 0; temp[i] != '\0'; i++)
    {
        b[i] = temp[i];
    }
    b[strlen(temp)] = '\0';
}

int main()
{
    char a[100], b[100];
    cout << "Enter two strings: ";
    cin >> a >> b;
    cout << "Before swapping: a = " << a << ", b = " << b << endl;
    swapString(a, b);
    cout << "After swapping: a = " << a << ", b = " << b << endl;

    return 0;
}
```

10.Write a function that takes an integer and returns the factorial of that number. Declare function parameter as read only reference. Call the function with some argument from main function, store the result and print it.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

void factorial(const int &n)
{
    int fact = 1;
    for (int i = 1; i <= n; i++)
    {
        fact *= i;
    }
    cout << "Factorial of " << n << " is " << fact << endl;
}

int main()
{
```

```
    int n;
    cout << "Enter a number: ";
    cin >> n;
    factorial(n);

    return 0;
}
```

11. Write a function `Strcpy` to copy one string to another with suitable formal parameter declarations. The following points must be considered:

- **Source string** must not get modified.
- **Target string** is allowed to get modified.
- The **pointers must be constant pointers**.

Use it to copy some strings.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

void Strcpy(const string &s1, string &s2)
{
    s2 = s1;
}

int main()
{
    string s1, s2;
    cout << "Enter a string: ";
    cin >> s1;
    Strcpy(s1, s2);
    cout << "Copied string: " << s2 << endl;
    return 0;
}
```

12. Write an inline function `add()` that takes three integer arguments and returns the sum of these arguments.

```cpp
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

inline void add(int &a, int &b, int &c)
{
    cout << a + b + c << endl;
}

int main()
{
    int a, b, c;
    cin >> a >> b >> c;
    add(a, b, c);

    return 0;
}
```

13. Consider the following two scenarios:

- **a)** We want to find out the maximum between three integers.
- **b)** We also want to find out the maximum element of an array of integers.

Write two overloaded functions for these two scenarios.

```cpp
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

void maxElement(vi arr)
{
    int max = INT_MIN;
    for (int i = 0; i < arr.size(); i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
    cout << "Max element: " << max << endl;
}
```

```
void maxElement(int a, int b, int c)
{
    int max = INT_MIN;
    if (a > max)
    {
        max = a;
    }
    if (b > max)
    {
        max = b;
    }
    if (c > max)
    {
        max = c;
    }
    cout << "Max element: " << max << endl;
}

int main()
{
    vi arr = {1, 2, 3, 4, 5};
    maxElement(arr);
    maxElement(1, 2, 3);

    return 0;
}
```

14. Write two overloaded functions `print()` such that one prints the elements of a vector and the other prints elements of a matrix. Note that a vector and a matrix may be represented as a one-dimensional array and a two-dimensional array respectively.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

void print(vector<vector<int>> arr)
{
    for (int i = 0; i < arr.size(); i++)
    {
        for (int j = 0; j < arr[i].size(); j++)
        {
```

```
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}

void print(vector<int> arr)
{
    for (int i = 0; i < arr.size(); i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{
    vector<vector<int>> arr = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    print(arr);
    vector<int> arr2 = {1, 2, 3, 4, 5};
    print(arr2);

    return 0;
}
```

15.Consider function `add()` in 13. Specify the default values for second and third parameters to 0 (zero). Now call this function with three, two and one arguments and see the result.

```
#include <bits/stdc++.h>
#define ll long long
#define vll vector<long long>
#define db double
#define vi vector<int>
using namespace std;

inline void add(int &a, int b = 0, int c = 0)
{
    cout << a + b + c << endl;
}

int main()
{
    int a = 1, b = 2, c = 3;
    add(a);
```

```
    return 0;
}
```