Homework

- Algorithm & Flowchart to find sum of series $1 - X + X^2 - X^3 + \cdots + X^N$. Where X and N are user inputs.

Step-1 Start

Step-2 Input Value of N, X

Step-3 I = 1, SUM=1, TERM=1

Step-4 IF (I >N) THEN
GO TO Step-9
ENDIF

Step-5 TERM = - TERM * X

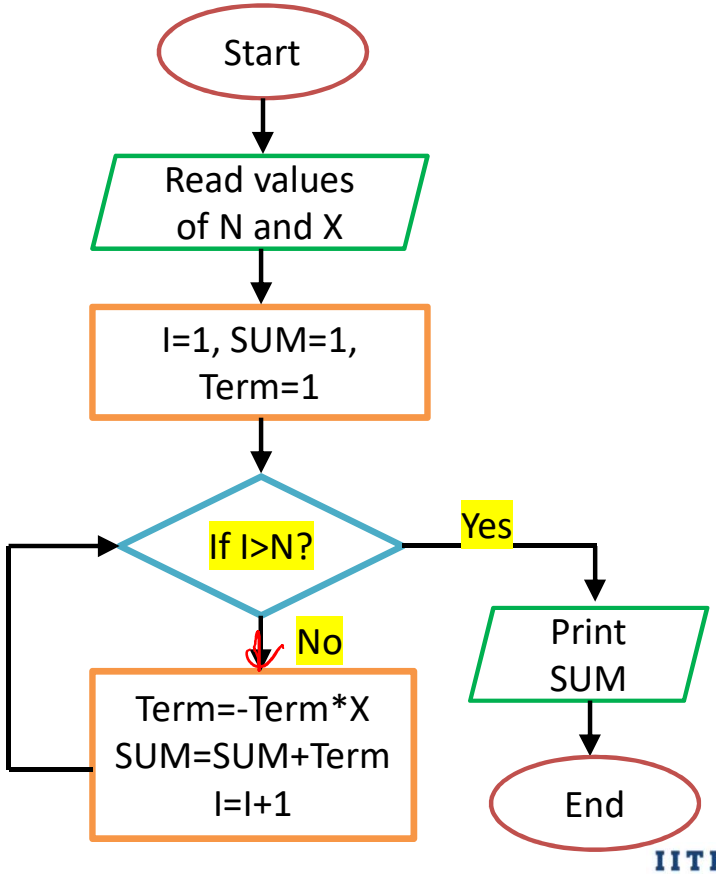
Step-6 SUM = SUM + TERM

Step-7 I = I + 1

Step-8 Go to step-4

Step-9 Display value of SUM

Step-10 Stop



INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

EEC-101
Programming with C++

Structure of a C++ program and input/output

Structure of a C++ Program

```
/*This is first C++ program
 It displays "Hello World!" on the screen
*/
#include <iostream> // preprocessor directives
using namespace std;

int main ()          // This is the main function
{
   cout << "Hello World!"<<endl; // print hello world to the screen

    return (0);
}
```

Components of a C++ Program

1. Comments
2. Keyword
3. Identifiers
4. Operators
5. Punctuation
6. Syntax

```
/*This is first C++ program
 It displays "Hello World!" on the
*/
#include <iostream> // preprocesso
using namespace std;

int main ()          // This is the
{
    cout << "Hello World!"<<endl;

     return (0);
}
```

Comments

- Comments are programmer-readable explanations in the source code.
- Comments are parts of the source code disregarded by the compiler. They simply do nothing.
- Their purpose is only to allow the programmer to insert notes or descriptions embedded within the source code.

C++ supports two ways to insert comments:

1. *// line comment*
2. *```
/* block comment
   line-1
   line-2
*/
```*
```

Comments

```
/*This is first C++ program
 It displays "Hello World!" on the screen
*/
#include <iostream> // preprocessor directives

using namespace std;
int main ()          // This is the main function
{
    cout << "Hello World!"<<endl; // print hello world to the screen

    return (0);
}
```

- All lines beginning with two slash signs (//) are considered comments and do not have any effect on the behavior of the program.
- The programmer can use them to include short explanations or observations within the source code itself.

C++ Keywords

```
1  // This is first C++ program
2  // It displays "Hello World!" on the screen
3
4  #include <iostream>
5  #include <string>
6  using namespace std;
7  int main ()
8  {
9      cout << "Hello World!"<<endl;
10
11     return (0);
12 }
```

- Keywords are part of the vocabulary of a programming language
- In most languages, the keywords are also reserved.
- This means that you, the programmer, cannot redefine their meaning, and you can't use them in a way in which they were not intended.

Identifiers

```
1   // This is first C++ program
2   // It displays "Hello World!" on the screen
3
4   #include <iostream>
5   #include <string>
6   using namespace std;
7   int main ()
8   {
9       cout << "Hello World!"<<endl;
10
11       return (0);
12  }
```

- Within a program names are used to designate variables and functions.
- Identifiers are programmer-defined names.
- Rules for naming identifiers will be discussed later

Operators

```
1  // This is first C++ program
2  // It displays "Hello World!" on the screen
3
4  #include <iostream>
5  #include <string>
6  using namespace std;
7  int main ()
8  {
9      cout << "Hello World!"<<endl;
10
11     return (0);
12 }
```

- Arithmetic operators (+,-, *)
- String resolution operator <<
- We will cover operators in details later

Punctuations

```
1   // This is first C++ program
2   // It displays "Hello World!" on the screen
3
4   #include <iostream>
5   #include <string>
6   using namespace std;
7   int main ()
8   {
9       cout << "Hello World!"<<endl;
10
11      return (0);
12  }
```

- Special characters that separate, and terminate items

Syntax

- Grammar of the programming language
- How programming elements are put together to form a program
- All programming language have rules

Preprocessor Directives

- What is preprocessor?
- It is a program that processes the source code before the compiler sees it.
  - Removes all comments and replace them with a single space
  - insert files,
  - apply machine-specific rules

```
#include <iostream>
#include "myfile.h"

#if
#elif
#else
#endif
```

Preprocessor Directives

#include <iostream>

- Lines beginning with a hash sign (#) are directives for the preprocessor. They are indications for the compiler's preprocessor.
- Directive #include <iostream> tells the preprocessor to include the iostream file.
- Angle brackets < > are required to indicate that the word iostream (which stands for input/output stream) is the name of a Standard C++ Library file
- The expression <iostream> is called standard header

Preprocessor

- The preprocessor performs preliminary operations on C++ files before they are passed to the compiler.
- Preprocessor can be used to conditionally compile code, insert files, specify compile-time error messages, and apply machine-specific rules to sections of code.

- #include <iostream> is a preprocessor directive that specifies the C++ compiler where to find the definitions of cout or cin in the **namespace std**

Preprocessor Directives

- The C++ preprocessor does not understand C++.

- It simply follows the preprocessor directives and gets the source code ready for the compiler.

The main() Function

```
4   #include <iostream>
5   #include <string>
6   using namespace std;
7   int main ()
8   {
9       cout << "Hello World!"<<endl;
10
11      return (0);
12  }
```

- This line corresponds to the beginning of the definition of the main function.
- The main function is the point by where all C++ programs start their execution, independently of its location within the source code.
- It does not matter whether there are other functions with other names defined before or after it - the instructions contained within this **i.e. main()** function's definition will always be the first ones to be executed in any C++ program.
- All C++ programs have a *main* function.

- Identifier main is the name of a function, called the main function of the program
- Every C++ program must have one and only one main () function
- The parentheses () indicate that it is function
- The keyword int is the name of the data type integer
- It is used to indicate the return type for main() function

- The word main is followed in the code by a pair of parentheses ().
- That is because it is a function declaration: In C++, what differentiates a function declaration from other types of expressions are these parentheses that follow its name.
- Optionally, these parentheses may enclose a list of parameters within them.

- Right after these parentheses of main() the body of the main function is enclosed in braces { }

- Whatever is contained within these { } braces is what the function does when it is executed

Namespace

```
#include <iostream> // preprocessor directives


int main ()          // This is the main function
{
    std::cout << "Hello World!"<<std::endl; // print hello world

     return (0);
}
```

- As our c++ programs become more and more complex, our programs become a combination of our own code, the c++ standard libraries and their code and libraries from third-party developers and their code.
- So as you can imagine, sooner or later we're going to run into the situation where company x names something the same as company y.
- When we use that name in our program, the c++ compiler doesn't know which one to use. So we have something called the naming conflict.

- A namespace is a named group of definitions
- The cout or cin object is defined within a namespace named std (for “standard”) in the <iostream> header file
- Namespace makes it possible for a program to use different objects with the same name, just as different people can have the same name

```cpp
#include <iostream> // preprocessor directives


int main ()          // This is the main function
{
    std::cout << "Hello World!"<<std::endl; // print hello world

     return (0);
}
```

```cpp
#include <iostream> // preprocess

using namespace std;
int main ()          // This is th
{
    cout << "Hello World!"<<endl;

    return (0);
}
```

Namespaces

- **using namespace** std;

- All the elements of the standard C++ library are declared within what is called a namespace, the namespace with the name *std*

- So in order to access its functionality we declare with this expression that we will be using these entities

- This line is very frequent in C++ programs that use the standard library

Basic Input/ Output

- cout << "Hello World!";
  This line is a C++ statement.
- A statement is a simple or compound expression that can actually produce some effect.
- This statement performs the only action that generates a visible effect in our first program.

  cout is declared in the *iostream* standard file within the *std* namespace

- we needed to include that specific file and to declare that we were going to use this specific namespace earlier in our code.

Semicolon

- All statements end with a semicolon character (;)
- This character is used to mark the end of the statement.
- It must be included at the end of all expression statements in all C++ programs

One of the most common syntax errors is indeed to forget to include some semicolon after a statement

- Preprocessor directives (those that begin by #) are out of this general rule since they are not statements

- They are lines read and processed by the preprocessor and do not produce any code by themselves

- Preprocessor directives must be specified in their own line and do not have to end with a semicolon (;).

Whitespace

Return Statement

- Terminates the execution of a function and returns control to the calling function
- In the case of the main function, transfers control back to the operating system
- Execution resumes in the calling function at the point immediately following the call.

- When the program has finished running, it can return an integer value to the operating system to signal some resulting status
- The return statement causes the main function to finish
- Return may be followed by a return code

- A return code of *0* for the *main* function is generally interpreted as the program worked as expected without any errors during its execution

- This is the most usual way to end a C++ console program

General Structure

```
preprocessing directives
namespace
    int main()
    {
    declarations;
    statements;
    }
```

Basic IO using Cin and Cout

- cout and cin, are defined in the c++ standard. To use them, you must include iostream.
- C++ uses a stream abstraction to handle IO on devices like the console and keyboard.

- cout is an output stream that defaults to the console or the screen.
- And cin is an input stream that defaults to the keyboard.

- The insertion operator(<<) is used with output streams.
- And the extraction operator (>>) is used with input streams

cout and <<

- Insert the data into the cout stream

```
cout << data;
```

- Can be chained

```
cout << "data 1 is " << data1;
```

- Does not automatically add line breaks

```
cout << "data 1 is " << data1 << endl;
cout << "data 1 is " << data1 << "\n";
```

cin and >>

- Extract data from the cin stream based on data's type

```
cin >> data;
```

- Can be chained

```
cin >> data1 >> data2;
```

- Can fail if the entered data cannot be interpreted

`data` could have an undetermined value

- If you are asked you to retain the number 5 in your mental memory, and then you are asked to memorize also the number 2 at the same time.

- You have just stored two different values in your memory. Now, if you are asked to add 1 to the first number (i.e. 5), you should be retaining the numbers 6 (that is 5+1) and 2 in your memory.

- The whole process that you have just done with your mental memory is similar to what a computer can do with two variables. The same process can be expressed in C++ with the following instruction set:
- a = 5;
- b = 2;
- a = a + 1;
- result = a - b;

(in this example we have yet not used the data types)

- Obviously, this is a very simple example since we have only used two small integer values, but consider that your computer can store millions of numbers like these at the same time and conduct sophisticated mathematical operations with them.

Homework

- Write a C++ program that asks the user to enter two integers. And display their sum on the screen.

Thanks