Lab Sheet 3

Operating Systems

C Programming

C Programming VS C++ Programming

C	C++
C was developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs.	C++ was developed by Bjarne Stroustrup in 1979 with C++'s predecessor "C with Classes".
When compared to C++, C is a subset of C++.	C++ is a superset of C. C++ can run most of C code while C cannot run C++ code.
C supports procedural programming paradigm for code development.	C++ supports both procedural and object oriented programming paradigms; therefore C++ is also called a hybrid language.
C does not support object oriented programming; therefore it has no support for polymorphism, encapsulation, and inheritance.	Being an object oriented programming language C++ supports polymorphism, encapsulation, and inheritance.
In C (because it is a procedural programming language), data and functions are separate and free entities.	In C++ (when it is used as object oriented programming language), data and functions are encapsulated together in form of an object. For creating objects class provides a blueprint of structure of the object.
In C, data are free entities and can be manipulated by outside code. This is because C does not support information hiding.	In C++, Encapsulation hides the data to ensure that data structures and operators are used as intended.
C, being a procedural programming, it is a function driven language.	While, C++, being an object oriented programming, it is an object driven language.
C does not support function and operator overloading.	C++ supports both function and operator overloading.
C does not allow functions to be defined inside structures.	In C++, functions can be used inside a structure.
C does not have namespace feature.	C++ uses NAMESPACE which avoid name collisions.

	A namespace is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it. Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when your code base includes multiple libraries. All identifiers at namespace scope are visible to one another without qualification. Identifiers outside the namespace can access the members by using the fully qualified name for each identifier.
C uses functions for input/output. For example scanf andprintf.	C++ uses objects for input output. For example cin and cout.
C does not support reference variables.	C++ supports reference variables.
C has no support for virtual and friend functions.	C++ supports virtual and friend functions.
C provides malloc() and calloc() functions for dynamic memory allocation, and free() for memory de-allocation.	C++ provides new operator for memory allocation and deleteoperator for memory de-allocation.
C does not provide direct support for error handling (also called exception handling)	C++ provides support for exception handling. Exceptions are used for "hard" errors that make the code incorrect.

What do I need to know to start out with C programming?

- **Library to be used**: #include <stdio.h> // standard input and output

- **Input and output commands:**
  1. **Printf("")** – this only prints out strings, in order to print out other variables such as int or floats it will be in this form:
     Int x=5; float y=6.2;
     Printf("%d and %f",x,y); // %d defines decimal numbers, %f defines floating numbers.
  2. **Scanf_s**("type of variable", &variable) – this scans any incoming input based on definition e.g int x; scanf_s("%d",&x);
  3. To read strings we need to create an array of characters (original string format) and use **get_s**(character array). Recommended size of array 256. To get size of inputted string use _mbstrlen(array);
  4. After reading any integer or numeric number, there is a trailing number character that the buffer waits for to complete the function. This will not affect your operations in any way, but when coming to input a character value or a string after it will terminate before you can continue or delay a trailing command. To avoid this issue add **getchar();** after a numeric input.
  5. When inputting a **character,** it is best to use getchar();
- **IDE** – Visual Studio, create a c++ project

The rest of c is similar to c++ programming in terms of conditional statements, iterative statements or even functions. As most people will say c is a subset of c++, hence if you have experience in c++ you will have no trouble with c.

Problem Set

Problem 1: **Searching 101**

Write a program that allows the user to enter exactly 5 numbers as initial input.
The program must then ask for an additional number.

The output of the program will indicate whether the last number is contained in the first 5 numbers.

Your output should resemble that provided in the example below.

All of the numbers that will be entered are integers.

Example 1

Enter a number: 1
Enter a number: 2
Enter a number: 5
Enter a number: 10
Enter a number: 15
Enter the number to be searched: 10

The number 10 appears in the first 5 numbers.

Example 2

Enter a number: 10
Enter a number: 20
Enter a number: 30
Enter a number: 40
Enter a number: 50
Enter the number to be searched: 60

The number 60 does not appear in the first 5 numbers.

Problem 2: **Do You Have Enough Money?**

General Statement:
Read the amount of money you have and the prices of the items you intend to buy. Determine whether you have enough money to buy everything you selected or whether you are short of money. If you do not have enough money, indicate the amount of the shortfall. Be sure to include 8% tax when figuring the amount you need.

Input:
The first line in the data set is an integer that represents the number of data collections that follow. There are an unknown number of money amounts in each data set (first input is the amount of money you have). The value –1 is used to indicate the end of the collection of prices.

Output:
All letters are to be upper case. Include the amount of shortfall if you do not have enough money. This money amount is to have a dollar sign ($) in front of the amount and it is to be rounded to 2 decimal places. The output is to be formatted exactly like that for the sample output given below.

Assumptions:
The –1 used to indicate the end of a data collection is not part of the data for the problem.

Sample Input:
3
10.50 7.60 1.26 3.49 -1
15.75 6.00 3.98 -1
21.00 5.25 5.75 4.76 3.98 1.50 -1

Sample Output:
$2.84 SHORT
ENOUGH MONEY
$1.94 SHORT

Problem 3: **Parallelogram Words**

General Statement:
Output a given word horizontally and multiple times vertically so that each letter in the horizontal word matches the position of that letter vertically.

Input:
The first line in the data set is an integer that represents the number of words that follow. Each word is on a separate line.

Output:
The horizontal word is to be in the center of the output. Each vertical word is to read down from the top. The first vertical word uses the first letter of the horizontal word. The last vertical word uses the last letter of the horizontal word.

All outputs are to have the same left edge vertically.

The output is to be formatted exactly like that for the sample output given below.

Assumptions:
All letters are upper case. The maximum word length is 10.

Sample Input:
3
TEST
SAMPLE
ART

Sample output:

```
   T
  TE
 TES
TEST
EST
ST
T

     S
    SA
   SAM
  SAMP
 SAMPL
SAMPLE
AMPLE
MPLE
PLE
LE
E

  A
 AR
ART
RT
T
```