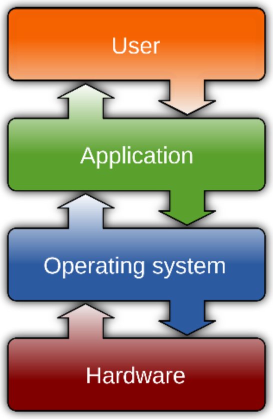
The
BRITISH
UNIVERSITY
IN EGYPT

Operating Systems – Lab#1

What is an Operating System?

An **Operating System** (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

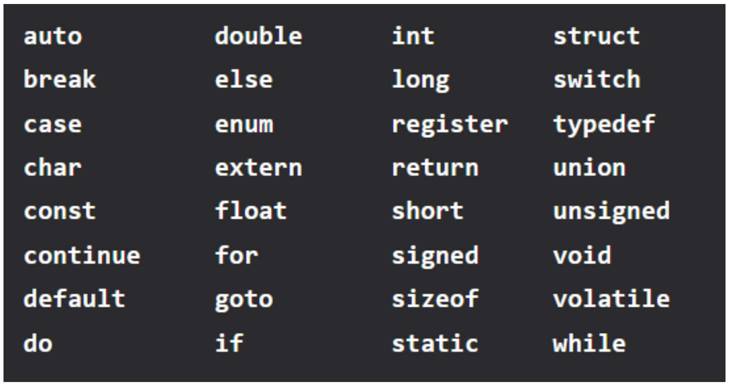


Differences between C and C++ programming languages:

C	C++
C is a subset of C++.	C++ is a superset of C.
C contains 32 keywords	C++ contains 63 keywords
C supports procedural programming	C++ is known as hybrid language as it supports both procedural and object-oriented programming paradigms
Header file used by C is stdio.h	Header file used by C++ is iostream.h
Virtual and friend functions are not supported by C.	Virtual and friend functions are supported by C++.
C does not support inheritance.	C++ supports inheritance.
C follows the top-down approach	C++ follows the Bottom-up approach
C does not support overloading	C++ does support overloading
scanf() and printf() functions are used for input/output in C.	cin() and cout() are used for input/output in c++

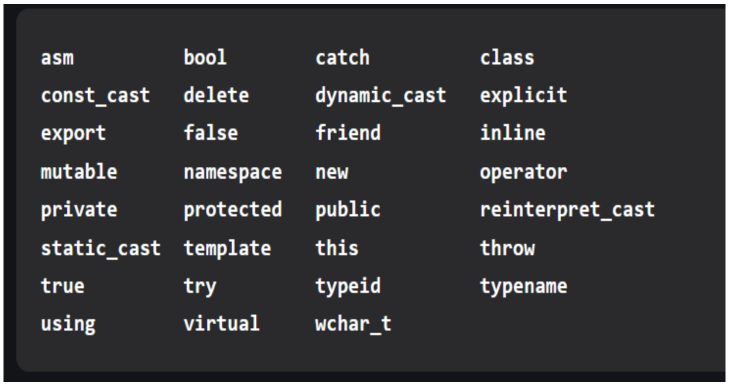
Operating Systems – Lab#1

C language supports **32** keywords which are given bellow:



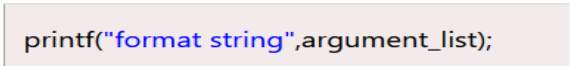
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

While in **C++** there are **31** additional keywords other than **C** Keywords they are:

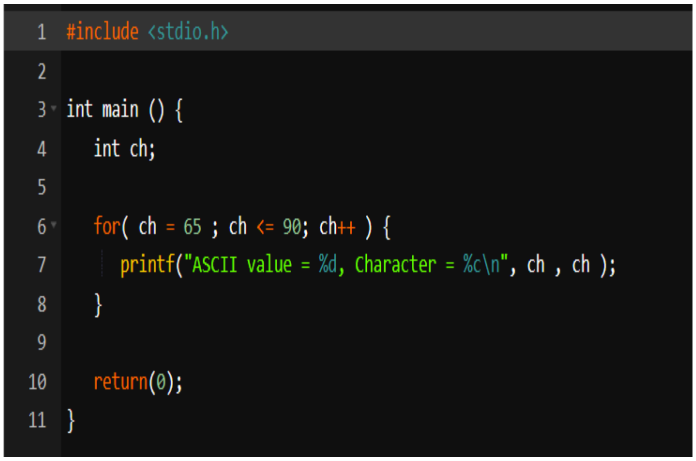


asm	bool	catch	class
const_cast	delete	dynamic_cast	explicit
export	false	friend	inline
mutable	namespace	new	operator
private	protected	public	reinterpret_cast
static_cast	template	this	throw
true	try	typeid	typename
using	virtual	wchar_t	

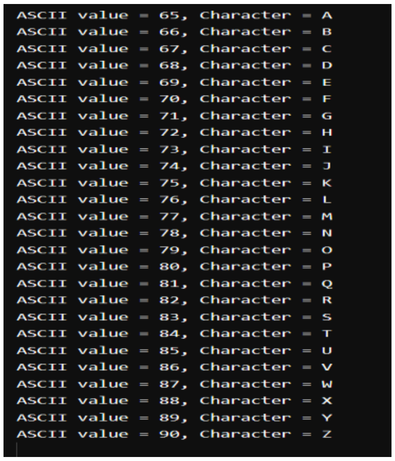
How to print data in C?



```
printf("format string",argument_list);
```



```
#include <stdio.h>

int main () {
   int ch;

   for( ch = 65 ; ch <= 90; ch++ ) {
      printf("ASCII value = %d, Character = %c\n", ch , ch );
   }

   return(0);
}
```



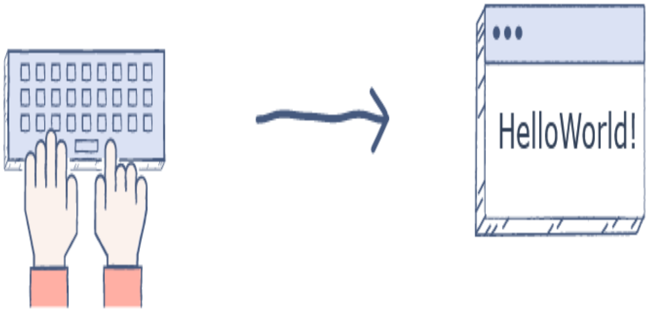
```
ASCII value = 65, Character = A
ASCII value = 66, Character = B
ASCII value = 67, Character = C
ASCII value = 68, Character = D
ASCII value = 69, Character = E
ASCII value = 70, Character = F
ASCII value = 71, Character = G
ASCII value = 72, Character = H
ASCII value = 73, Character = I
ASCII value = 74, Character = J
ASCII value = 75, Character = K
ASCII value = 76, Character = L
ASCII value = 77, Character = M
ASCII value = 78, Character = N
ASCII value = 79, Character = O
ASCII value = 80, Character = P
ASCII value = 81, Character = Q
ASCII value = 82, Character = R
ASCII value = 83, Character = S
ASCII value = 84, Character = T
ASCII value = 85, Character = U
ASCII value = 86, Character = V
ASCII value = 87, Character = W
ASCII value = 88, Character = X
ASCII value = 89, Character = Y
ASCII value = 90, Character = Z
```

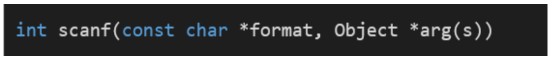
The BRITISH UNIVERSITY IN EGYPT

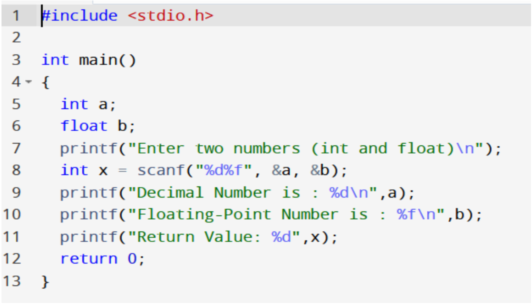
Operating Systems – Lab#1

How to read data in C?

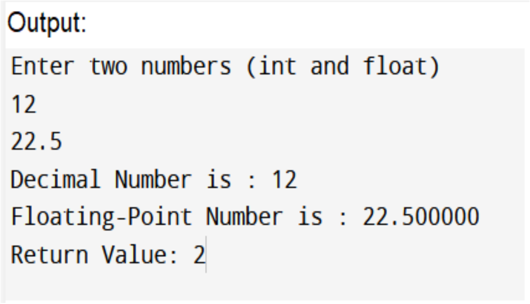
In C the scanf() function is used to read formatted data from the console.



Syntax	 `int scanf(const char *format, Object *arg(s))`
Parameters	• Object: Address of the variable(s) which will store data • char *: This contains the *format specifiers*
Format specifier (special character which is used to specify the data type of the value being read)	• **s -** strings • **d** – decimal integers • **f** – floating-point numbers • **c** – a single character
Return value	• If the function successfully reads the data, the number of items read is returned • In case of unsuccessful execution, a negative number is returned • If there is an input failure, EOF is returned



```
#include <stdio.h>

int main()
{
  int a;
  float b;
  printf("Enter two numbers (int and float)\n");
  int x = scanf("%d%f", &a, &b);
  printf("Decimal Number is : %d\n",a);
  printf("Floating-Point Number is : %f\n",b);
  printf("Return Value: %d",x);
  return 0;
}
```

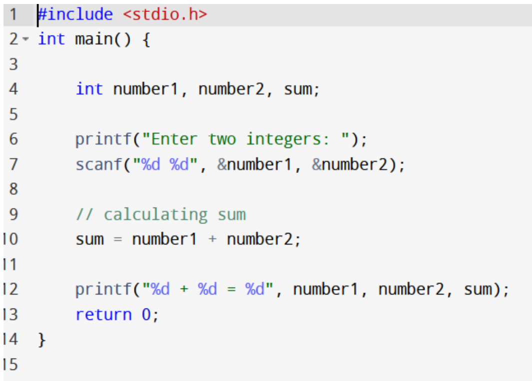


Output:

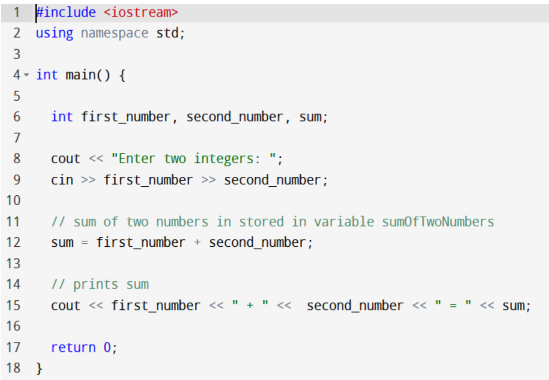
```
Enter two numbers (int and float)
12
22.5
Decimal Number is : 12
Floating-Point Number is : 22.500000
Return Value: 2
```

Operating Systems – Lab#1

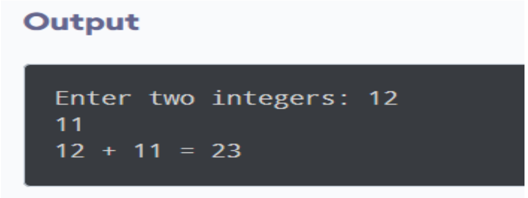
Comparison (C and C++): Write a program to add two integers.



```c
#include <stdio.h>
int main() {

    int number1, number2, sum;

    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    // calculating sum
    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}

```



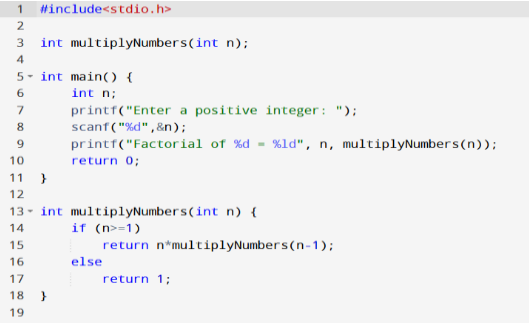
```cpp
#include <iostream>
using namespace std;

int main() {

    int first_number, second_number, sum;

    cout << "Enter two integers: ";
    cin >> first_number >> second_number;

    // sum of two numbers in stored in variable sumOfTwoNumbers
    sum = first_number + second_number;

    // prints sum
    cout << first_number << " + " <<  second_number << " = " << sum;

    return 0;
}
```



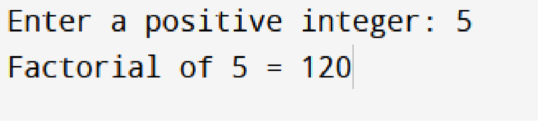
Output

```
Enter two integers: 12
11
12 + 11 = 23
```

Exercise 1: Write a program that calculates the factorial of a number using recursion.

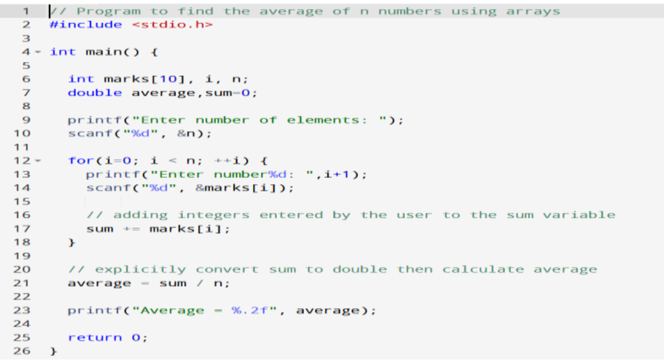


```c
#include<stdio.h>

int multiplyNumbers(int n);

int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}

```

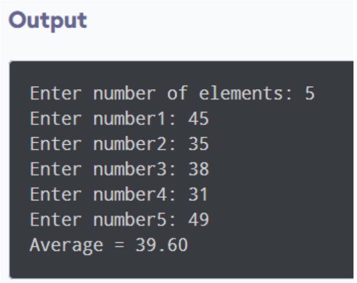


```
Enter a positive integer: 5
Factorial of 5 = 120
```

Exercise 2: Write a program to find the average of n numbers using arrays.



```c
// Program to find the average of n numbers using arrays
#include <stdio.h>

int main() {

   int marks[10], i, n;
   double average,sum=0;

   printf("Enter number of elements: ");
   scanf("%d", &n);

   for(i=0; i < n; ++i) {
     printf("Enter number%d: ",i+1);
     scanf("%d", &marks[i]);

     // adding integers entered by the user to the sum variable
     sum += marks[i];
   }

   // explicitly convert sum to double then calculate average
   average = sum / n;

   printf("Average = %.2f", average);

   return 0;
}
```



Output

```
Enter number of elements: 5
Enter number1: 45
Enter number2: 35
Enter number3: 38
Enter number4: 31
Enter number5: 49
Average = 39.60
```