<https://www.onlinegdb.com/online_c_compiler>

C Programming Language

The C Language is developed for creating system applications that direct interacts to the hardware devices such as drivers, kernels etc.

C programming is considered as the base for other programming languages, that is why it is known as mother language.

It can be defined in the following ways:

1. Mother language

2. System programming language

3. Procedure-oriented programming language

4. Structured programming language

5. Mid-level programming language

1-C language is considered as the mother language of all the modern languages because of most of the compilers, JVMs, Kernals etc. are written in C language and most of the languages follow c syntax e.g. C++, Java etc.

It provides the core concepts like array, functions, file handling etc. that is being used in many languages like C++, Java, C# etc.

2) C as a system programming language

A system programming language is used to create system software. C language is a system programming language because it can be used to do low-level programming (e.g. driver and kernel). It is generally used to create hardware devices, OS, drivers, kernels etc. For example, the Linux kernel is written in C.

It can be used in internet programming like java, .net, PHP etc.

3) C as a procedural language

A procedure is known as function, method, routine, subroutine etc. A procedural language specifies a series of steps or procedures for the program to solve the problem.

A procedural language breaks the program into functions, data structures etc.

C is a procedural language. In C, variables and function prototypes must be declared before being used.

4) C as a structured programming language

A structured programming language is a subset of procedural language. Structure means to break a program into parts or blocks so that it may be easy to understand.

In C language, we break the program into parts using functions. It makes the program easier to understand and modify. C programming language was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A.

Dennis Ritchie is known as the founder of c language.

**Features of C Language-**

C is the widely used language. It provides a lot of featuresthat are given below.

1. Simple

2. Machine Independent or Portable

3. Mid-level programming language

4. structured programming language

5. Rich Library

6. Memory Management

7. Fast Speed

8. Pointers

9. Recursion

10. Extensible

1) Simple

C is a simple language in the sense that it provides structured approach (to break the problem into parts), rich set of library functions, data types etc.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) Machine Independent or Portable

Unlike assembly language, c programs can be executed in many machines with little bit or no change. But it is not platform-independent.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) Mid-level prorgramming language

C is also used to do low level programming. It is used to develop system applications such as kernel, driver etc. It also supports the feature of high level language. That is why it is known as mid-level language.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) Structured prorgramming language

C is a structured programming language in the sense that we can break the program into parts using functions. So, it is easy to understand and modify.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) Rich Library

C provides a lot of inbuilt functions that makes the development fast.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6) Memory Management

It supports the feature of dynamic memory allocation. In C language, we can free the allocated memory at any time by calling the free() function.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7) Speed

The compilation and execution time of C language is fast.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8) Pointer

C provides the feature of pointers. We can directly interact with the memory by using the pointers. We can use pointers for memory, structures, functions, array etc.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9) Recursion

In c, we can call the function within the function. It provides code reusability for every function.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10) Extensible

C language is extensible because it can easily adopt new features.

**Program to print cube of given number**

Let's see a simple example of c language that gets input from the user and prints the cube of the given number.

#include<stdio.h>

#include<conio.h>

void main(){

int number;

printf("enter a number:");

scanf("%d",&number);

printf("cube of number is:%d ",number\*number\*number);

getch();

}

**the flow of the program-**

1) C program (source code) is sent to preprocessor first. The preprocessor is responsible to convert preprocessor directives into their respective values.

The preprocessor generates expanded source code.

2) Expanded source code is sent to the compiler which compiles the code and converts it into assembly code.

3) The assembly code is sent to the assembler which assembles the code and converts it into object code. Now a simple.obj file is generated.

4) The object code is sent to the linker which links it to the library such as header files. Then it is converted into executable code. A simple.exe file is generated.

5) The executable code is sent to loader which loads it into memory and then it is executed. After execution, the output is sent to console.

**printf scanf in C-**

The printf() and scanf() functions are used for input and output in C language. Both functions are inbuilt library functions, defined in stdio.h (header file).

printf() function

The printf() function is used for output. It prints the given statement to the console.

The syntax of printf() function is given below:

printf("format string",argument\_list);

The format string can be %d (integer), %c (character), %s (string), %f (float) etc.

scanf() function

The scanf() function is used for input. It reads the input data from the console.

scanf("format string",argument\_list);

**Variables in C**

A variable is the name of the memory location. It is used to store data. Its value can be changed and it can be reused many times.

It is a way to represent memory location through a symbol so that it can be easily identified.

Let's see the syntax to declare a variable:

1. type variable\_list;

The example of declaring variable is given below:

1. int a;

2. float b;

3. char c;

Here, a, b, c are variables and int,float,char are data types.

We can also provide values while declaring the variables as given below:

1. int a=10,b=20;//declaring 2 variable of integer type

2. float f=20.8;

3. char c='A';

**Rules for defining variables-**

o A variable can have alphabets, digits and underscore.

o A variable name can start with the alphabet and underscore only. It can't start with a digit.

o No white space is allowed within the variable name.

o A variable name must not be any reserved word or keyword e.g. int, float etc.

Valid variable names:

1. int a;

2. int \_ab;

3. int a30;

Inalid variable names:

1. int 2;

2. int a b;

3. int long;

**Types of Variables in C**

There are many types of variables in c:

1. local variable

2. global variable

3. static variable

4. automatic variable

5. external variable

Local Variable

A variable that is declared inside the function or block is called a local variable.

It must be declared at the start of the block.

1. void function1(){

2. int x=10;//local variable

3. }

You must have to initialize the local variable before it is used.

Global Variable

A variable that is declared outside the function or block is called global variable. Any function can change the value of the global variable. It is available to all the functions.

It must be declared at the start of the block.

1. int value=20;//global variable

2. void function1(){

3. int x=10;//local variable

4. }

Static Variable

A variable that is declared with static keyword is called static variable.

It retains its value between multiple function calls.

1. void function1(){

2. int x=10;//local variable

3. static int y=10;//static variable

4. x=x+1;

5. y=y+1;

6. printf("%d,%d",x,y);

7. }

If you call this function many times, local variable will print the same value for each function call e.g, 11,11,11 and so on. But static variable will print the incremented value in each function call e.g. 11, 12, 13 and so on.

Automatic Variable

All variables in C that is declared inside the block, are automatic variables by default. By we can explicitly declare automatic variable using auto keyword.

1. void main(){

2. int x=10;//local variable (also automatic)

3. auto int y=20;//automatic variable

4. }

External Variable

We can share a variable in multiple C source files by using external variable. To declare a external variable, you need to useextern keyword.

myfile.h

1. extern int x=10;//external variable (also global)

program1.c

1. #include "myfile.h"

2. #include <stdio.h>

3. void printValue(){

4. printf("Global variable: %d", global\_variable);

5. }

**Data Types in C**

A data type specifies the type of data that a variable can store such as integer, floating, character etc.

There are 4 types of data types in C language.

Type Data Types

Basic Data Type - int, char, float, double

Derived Data Type- array, pointer, structure, union

Enumeration Data Type- enum

Void Data Type - void

Basic Data Types

The basic data types are integer-based and floating-point based. C language supports both signed and unsigned literals.

The memory size of basic data types may change according to 32 or 64-bit operating system.

Let's see the basic data types. Its size is given according to 32 bit OS.

Data Types Memory Size Range

Char 1 byte -128 to 127

signed char 1 byte -128 to 127

unsigned char 1 byte 0 to 127

Short 2 byte -32,768 to 32,767

signed short 2 byte -32,768 to 32,767

unsigned short 2 byte 0 to 32,767

Int 2 byte -32,768 to 32,767

signed int 2 byte -32,768 to 32,767

unsigned int 2 byte 0 to 32,767

short int 2 byte -32,768 to 32,767

signed short int 2 byte -32,768 to 32,767

unsigned short int 2 byte 0 to 32,767

long int 4 byte

signed long int 4 byte

unsigned long int 4 byte

Float 4 byte

Double 8 byte

long double 10 byte

**Keywords in C**

A keyword is a reserved word. You cannot use it as a variable name, constant name etc. There are only 32 reserved words (keywords) in C language.

A list of 32 keywords in c language is given below:

Auto break case char const continue default do

Double else enum extern float for goto if

Int long register return short s

**C Operators**

An operator is simply a symbol that is used to perform operations. There can be many types of operations like arithmetic, logical, bitwise etc.

There are following types of operators to perform different types of operations in C language.

o Arithmetic Operators

o Relational Operators

o Shift Operators

o Logical Operators

o Bitwise Operators

o Ternary or Conditional Operators

o Assignment Operator

o Misc Operator

**Precedence of Operators in C**

The precedence of operator species that which operator will be evaluated first and next. The associativity specifies the operator's direction to be evaluated, it may be left to right or right to left.

Let's understand the precedence by the example given below:

1. int value=10+20\*10;

The value variable will contain 210 because \* (multiplicative operator) is evaluated before + (additive operator).

The precedence and associativity of C operators is given below:

Category Operator Associativity

Postfix () [] -> . ++ - - Left to right

Unary + - ! ~ ++ - - (type)\* & sizeof Right to left

Multiplicative \* / % Left to right

Additive + - Left to right

Shift << >> Left to right

Relational < <= > >= Left to right

Equality == != Left to right

Bitwise AND & Left to right

Bitwise XOR ^ Left to right

Bitwise OR | Left to right

Logical AND && Left to right

Logical OR || Left to right

Conditional ?: Right to left

Assignment = += -= \*= /= %=>>= <<= &= ^= |= Right to left

Comma , Left to right

**Comments in C**

Comments in C language are used to provide information about lines of code. It is widely used for documenting code. There are 2 types of comments in C language.

1. Single-Line Comments //

2. Multi-Line Comments /\*

**Escape Sequence in C**

An escape sequence in C language is a sequence of characters that doesn't represent itself when used inside string literal or character.

It is composed of two or more characters starting with backslash \. For example, \n represents a new line.

List of Escape Sequences in C

Escape Sequence Meaning

\a Alarm or Beep

\b Backspace

\f Form Feed

\n New Line

\r Carriage Return

\t Tab (Horizontal)

\v Vertical Tab

\\ Backslash