**TYPES OF KEY WORDS IN C PROGRAMMING**

1. **AUTO KEYWORD**

* It is used to declare automatic variables that have a local scope within a block or function. For example: auto int x = 5;

**2. BREAK KEYWORD**

* It is used to terminate the execution of an innermost loop or switch statement and transfer control to the next statement after the loop or switch
* For example: for(int i = 0; i < 10; i++) {  
   if(i == 5) break;}
  1. **CASE**
* It is used within a switch statement to define specific cases for different values of the controlling expression.
* For example: switch(x) {  
   case 1:  
   // code  
   break;  
   case 2:  
   // code  
   break;}
  1. **CHAR**
* It is a data type used to represent a single character type variables
* .For example: char myChar = 'A';

1. **CONST**

* It is used to declare [constants](https://unstop.com/blog/constant-in-c), which are integer variables whose values cannot be modified once assigned. For example:

const int PI = 3.14;

1. **CONTINUE**

* It is used to skip the remaining statements within a loop iteration and start the next iteration
* For example: for(int i = 0; i < 10; i++) {  
  if(i == 5) continue;  
  // code  
  }

1. **DEFAULT**

* It is used within a switch statement to specify the default case when none of the other cases match:
* For example switch(x) {  
   case 1:  
   // code  
   break;  
   default:  
   // default code  
   }

1. **DO**

* It is used to create a do-while loop, which executes a block of code repeatedly until a specified condition is false. For example: do {  
  // code  
  } while(condition);

1. **DOUBLE**

* double is a data type used to represent double-precision floating-point numbers. For example: double myDouble = 3.14;

1. **ELSE**

* It is used to specify an alternative block of code to execute when the condition of an if statement is false. For example: if(condition) {  
  // code  
  } else {  
  // else code  
  }

1. **ENUM**

* It is used to define an enumeration, which is a set of named integer constants. For example: enum Days { MON, TUE, WED, THU, FRI, SAT, SUN }

1. **EXTERN**

* It is used to declare a variable or function that is defined in another source file. For example: extern int globalVariable;

1. **FLOAT**

* It is a data type used to represent single-precision floating-point numbers. For example: float myFloat = 2.5;

1. **FOR**

* It is used to create a [for loop](https://unstop.com/blog/for-loop-in-c), which executes a block of code repeatedly based on a specified initialization, condition, and [increment/decrement](https://unstop.com/blog/increment-and-decrement-operators-in-c). For example for(int i = 0; i < 10; i++) {code//}:

1. **GOTO**

* Goto statement is used to transfer control to a labeled statement within the same function. For example: goto label;  
  // code  
  label:  
  // code after goto

1. **IF**

* It is used to create an if statement, which executes a block of code based on a specified condition. For example if(condition) {  
  // code  
  }

1. **INT**

* It is a data type used to represent integers. For example: int myInt = 42;

1. **LONG**

* It is a data type used to represent long integers. For example: long myLong = 1000000L;

1. **REGISTER**

* It is used to suggest to the compiler to store the variable in a register for faster access, thus creating register variables. For example: register int regVar = 10;

1. **RETURN**

* It is used to return a value from a function. For example: return 0;

1. **SHORT**

* It is a data type used to represent short integers. For example: short myShort = 32767;

1. **SIGNED**

* It is a data type qualifier used to indicate that a variable can represent both positive and negative values and alter the meaning of a base data type to a new type. For example: signed int mySignedInt = -5;

1. **SIZEOF**

* It is used to determine the size in bytes of a data type or variable. It can also be used to return the size of the expression. For example: int size = sizeof(int);

1. **STATIC**

* It is used to declare static variables and functions with a local scope that retains their values across multiple function calls. For example: static int staticVar = 20;

1. **STRUCT**

* It is used to define a structure type, which is a collection of related variables. For example: struct Point {   
  int x;  
  int y;};

1. **SWITCH**

* It is used to create a switch statement (or switch case statement), which selects one of many code blocks to execute based on the value of an expression.

1. **TYPEDEF**

* It is used to create a new data type alias by providing an alternative name for an existing data type. For example: typedef unsigned int uint;

1. **UNION**

* It is used to define a union, which is a data structure that can store different types of data in the same memory location. For example: union Data {  
  int intValue;  
  float floatValue;  
  };

1. **UNSIGNED**

* It is a data type qualifier used to indicate that a variable can represent only positive values or zero. For example: unsigned int positiveInt = 10;

1. **VOID**

* It is used to specify that a function does not return a value or to declare generic pointers. For example: void myFunction() {  
  // code  
  }

1. **VOLATILE**

* It is used to indicate that a variable can be modified by external factors and should not be optimized by the compiler. For example: volatile int volatileVar = 5;

1. **WHILE**

* It is used to create a [while loop](https://unstop.com/blog/while-loop-in-c), which repeatedly executes a block of code as long as a specified condition is true. For example: while(condition) {  
  // code  
  }