Data type in C programming

In the C programming language, data types refer to the system extension used for variable declarations of different types. The type of variable determines the amount of memory used to store that variable and how the bits are stored when released

In the C programming language, data types refer to the system extension used for variable declarations of different types. The type of variable determines the amount of memory used to store that variable and how the bits are stored when interpreted.

The types of variables in C are divided as follows:

STT **Type and description1 Basic type**

Arithmetic types and include two main types: a) integer type and b) floating point real number type.

2 **Listing type**

These are arithmetic types and are used to define variables that can be assigned a certain number of integer values ??throughout the program.

3 **void type**

The *void* type is a special type that indicates that there is no value.

4 **Basic word development type**

Types include: a) pointer, b) array type, c) structure type, d) union type and e) function type (function).

Array and structure data types are used in collections as aggregate data types. Types are functions that specify the type of type the function returns. We will look at the basic data types in the following section, in which the remaining types will be mentioned in later chapters.

Integer type (type int) in C

The following table gives you a detailed understanding of the integer type with its storage size and its limits:

TypeArchiveRate of 1 byte -128 to 127 or 0 to 255 unsigned 1 byte char 0 to 255 signed char 1 byte -128 to 127 int 2 or 4 bytes -32,768 to 32,767 or -2,147,483,648 to 2,147,483,647 unsigned int 2 or 4 bytes 0 to 65,535 or 0 to 4,294,967,295 short 2 bytes -32,768 to 32,767 unsigned short 2 bytes 0 to 65,535 long 4 bytes -2,147,483,648 to 2,147,483,647 unsigned long 4 bytes 0 to 4,294,967,295

You can get the exact size of the types of variables on specific platforms, you can use the **sizeof** operator. The expression **sizeof (kieu)** returns the size of the object or the type in bytes. Below is an example to retrieve the size of the int object on any computer.

```
#include #include int  main () {  printf ( "Kich co luu tru cho so nguyen (int)
```

Compiling and running the above C program will result:

```
Kich  co luu tru cho so nguyen  ( int )  la : 4
```

Floating point type (Floating-Point) in C

The following table gives you a detailed understanding of standard floating-point number types with storage size and range and accuracy:

TypeStore storageData value Accuracy of 4 bytes 1.2E-38 to 3.4E + 38 6 8 decimal places double position 2.3E-308 to 1.7E + 308 15 10 byte double double decimal position 3.4E-4932 to 1.1 E + 4932 19 decimal places *float.h* in the Header file defines macros that allow you to use these values ??and other specific types of binary representation values ??of real numbers in your program. Below is an example that will print the size of the float type as well as its range of values:

```
#include
#include

int main ()
{
printf ("The storage class for real numbers (float) is:% dn", sizeof (float));
printf ("The minimum positive real value is:% En", FLT_MIN);
printf ("The largest positive real value is:% En", FLT_MAX);
printf ("Accuracy:% dn", FLT_DIG);

return 0;
}
```

Compiling and running the above C program will result:

```
Storage class for real numbers (float) is: 4
The smallest positive real value is: 1.175494E-38
The largest positive real value is: 3.402823E + 38
Accuracy: 6
```

Void type in C

Void type defines no value. It is used in the following 3 cases:

Function returns void: There are many functions in C language that do not return any data and you can say that it is a void function. A function that does not return any value is of type void. For example: void exit (int status);

Function with void parameter: There are functions in C that do not accept any parameters. A function with no acceptable parameters is a void. Example: int rand (void);

Cursor to void: A pointer has type void * representing the object's address, not a type. Example memory allocation function void * malloc (size_t size); returning a void pointer can cast to any object.

You may not understand these points in terms of void, we should continue and in the next chapters, we will repeat these points.

According to Tutorialspoint

Previous article: Basic syntax of C programming

Next lesson: Turning in programming C