Press type in C

Type is a way to convert a variable from one data type to another.

Type is a way to convert a variable from one data type to another. For example, when you want to store a long value for an integer, you must cast long to int. You can convert values ??from one type to another using the cast **operator** as follows:

```
( ten - kieu )  bieu_thuc
```

Consider the following example that the cast operator allows to split an integer variable to be executed as a floating-point operation:

```
#include  main () { int  sochia  = 32 ,  sobichia  = 6 ; double  kq ;  kq  = (
```

When executing the code, the following result is printed, the resulting variable has type double:

Compile and execute the above C program to see the results:

It should be kept in mind that the **cast** operator has a precedence over division, so the first **sochia** value is converted to a **double** and eventually it is divided by calculation in the double value field.

Type transforms can be hidden ie done automatically by the compiler, or it can be explicitly defined using a cast operator. It is good for you to use a cast operator anywhere that needs type conversion.

Integer upgrade in C

Integer upgrade is the process by which integer values ??smaller than **int** or **unsigned int** convert to type **int** or **unsigned int** . Suppose you have an example of adding a character to an int:

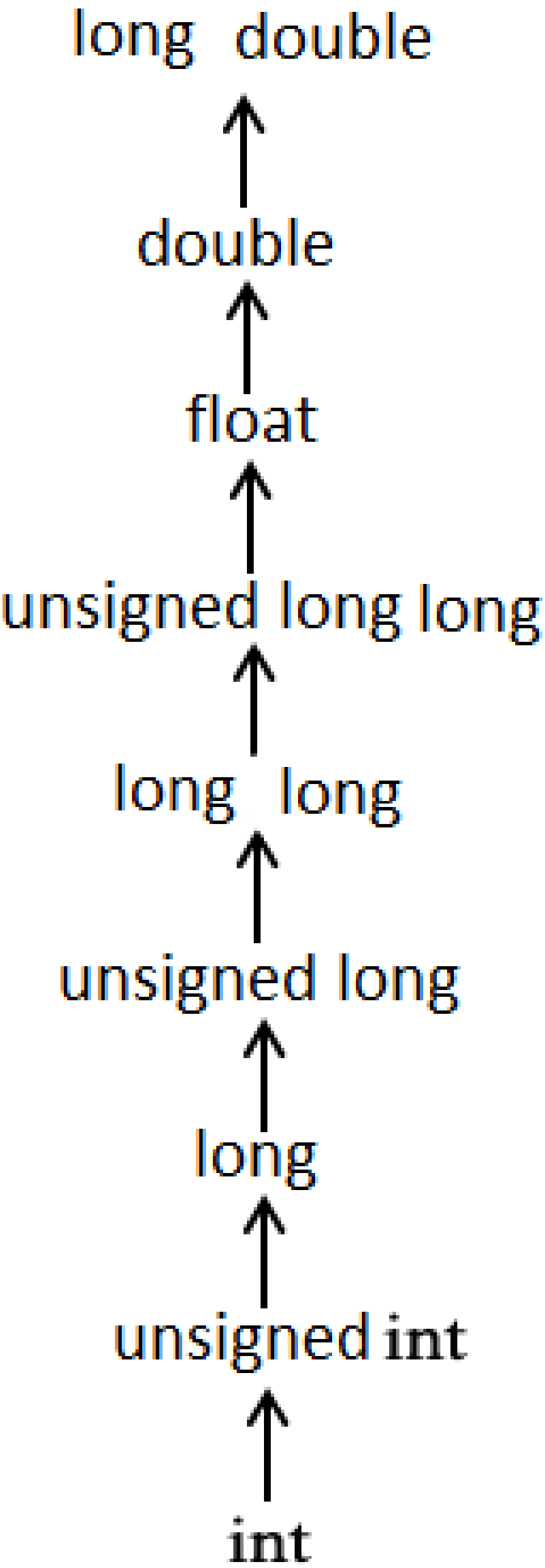
```
#include  main () { int  i  = 21 ; char  c  = 'c' ; /* Gia tri ASCII la 99 */ i
```

Compile and execute the above C program to see the results:

Here, the value of variable tong is 120 because the compiler performs an integer upgrade and converts the 'c' value to ACII before performing further operations.

Normal arithmetic transformation

The usual arithmetic transformation is the way to cast its value into a commonly used type. The first compiler will perform an integer upgrade, which converts from low to high, below the hierarchy:


long double
double
float
unsigned long long
long long
unsigned long
long
unsigned int
int

Normal arithmetic conversion is not performed for assignment operators, for logical operators: && and ||. We follow the following example to understand this concept:

```
#include  main () { int  i  = 21 ; char  c  = 'c' ; /* Gia tri ASCII la 99 */ f
```

Compile and execute the above C program to see the results:

Here, the simple way to understand is that first value c turns into an integer, but because the final value is double, so the usual arithmetic transformation applies and the compiler transforms i and c into type float and get the result of adding float type.

According to Tutorialspoint

Previous post: Header File in C

Next lesson: Handling errors in C