#include<iostream>

using namespace std;

int main()

{

int a=10;

int \*b,\*\*c;

b=&a;

c=&b;

cout<<"a value is ="<<a<<endl;

cout<<"a address is ="<<&a<<endl;

cout<<"b value is ="<<\*b<<endl;

cout<<"b address is ="<<&b<<endl;

cout<<"c value is ="<<\*\*c<<endl;

cout<<"c adress is ="<<&c<<endl;

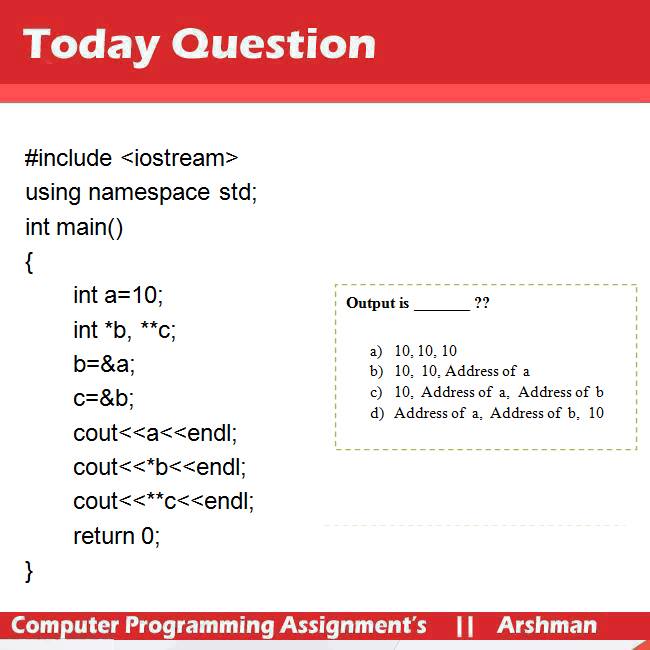
cout<<"a adress is ="<<&\*\*c<<endl;

getchar();

getchar();

return 0;

}



A pointer to a pointer is a form of multiple indirection or a chain of pointers. Normally, a pointer contains the address of a variable. When we define a pointer to a pointer, the first pointer contains the address of the second pointer, which points to the location that contains the actual value as shown below.



A variable that is a pointer to a pointer must be declared as such. This is done by placing an additional asterisk in front of its name. For example, following is the declaration to declare a pointer to a pointer of type int:

int \*\*var;