1. Explain the result of following program

#include <iostream>

using namespace std;

typedef int \*IntPtrType;

void main()

{

IntPtrType ptr\_a, ptr\_b, \*ptr\_c;//khoi tao con tro a, con tro b, con tro tro toi con tro c

ptr\_a = new int; //cap phat vung nho cho con tro a

\*ptr\_a = 3;//gan gia tri cho con tro a

ptr\_b = ptr\_a;//gan dia chia vung nho ma con tro a quan ly cho con tro b, tuc la a va b cung quan ly mot vung nho

cout << \*ptr\_a << " " << \*ptr\_b << "\n";//in ra gia tri ma con tro a va b cung quan ly

ptr\_b = new int;//cap phat vung nho moi cho con tro b

\*ptr\_b = 9;// gan gia tri cho con tro b

cout << \*ptr\_a << " " << \*ptr\_b << "\n";//in ra gia tri cua vung nho ma hai con tro tro den

\*ptr\_b = \*ptr\_a;// gan gia tri cua vung nho ma con tro a quan ly cho con tro b

cout << \*ptr\_a << " " << \*ptr\_b << "\n"; // in ra gia tri hai con tro do quan ly

delete ptr\_a;// thu hoi vung nho con tro a

ptr\_a = ptr\_b;// gan dia chi cua con tro b cho a

cout << \*ptr\_a << " " << \*ptr\_b << "\n";//in ra gia tri cua vung nho ma hai con tro a b cung tro den

ptr\_c = &ptr\_a; // gan dia chi cua con tro a cho con tro c, con tro tro den con tro

cout << \*ptr\_c << " " << \*\*ptr\_c << "\n";// in ra dia chi cua con tro a va gia tri cua dia chi ma con tro a tro den

delete ptr\_a;// thu hoi vung nho con tro a

ptr\_a = NULL;// gan lai bang NUll

system("pause");

}

1. Detect and solve problems of following program

#include <iostream>

using namespace std;

void main()

{

int a[4] = { 1, 2, 3, 4 };

int \*p = new int;

p = a;

int \*p2 = new int;

delete p;

delete[] a;

delete p2;

system("pause");

}

1. Using pointer:

* Why should we use delete?
* Để thu hồi lại vùng nhớ HEAP đã cấp phát.
* When we use delete?
* Khi sử dụng con trỏ cấp phát động
* Difference between delete and delete[]. Write a demo
* Delete dùng để giải phóng vùng nhớ của một biến độc lập ; còn delete [] thì giải phóng vùng nhớ của một mảng.

1. Given below code

\*p1 == 10

\*p2 == 0x200

\*(\*p2) == 10