I: POINTER:

1. Explain the result of following program.

#include <iostream>

using namespace std;

typedef int \*IntPtrType;

main()

{

IntPtrType ptr\_a, ptr\_b, \*ptr\_c; // khai báo 3 con trỏ

ptr\_a = new int; // cấp phát bộ nhợ con trỏ a, 4 byte

\*ptr\_a = 3; // gán giá tri con tro a = 3

ptr\_b = ptr\_a; //con trỏ b trỏ đến đại chỉ mà con trỏ b cũng trỏ tới.

cout << \*ptr\_a << " " << \*ptr\_b << "\n"; // 3 3

ptr\_b = new int; // cấp phát bộ nhớ cho con trỏ b, thay đổi đại chỉ mà con b trỏ tới

\*ptr\_b = 9; // gán giá trị con trỏ b = 9

cout << \*ptr\_a << " " << \*ptr\_b << "\n"; // 3 9

\*ptr\_b = \*ptr\_a; // giá trị con trỏ b = giá trị con trỏ a

cout << \*ptr\_a << " " << \*ptr\_b << "\n"; // 3 3

delete ptr\_a; // thu hồi vũng nhớ con trỏ a, con trỏ a vẫn trỏ tới địa chỉ củ nhưng không có giá trị.

ptr\_a = ptr\_b; // con trỏ a trỏ đến đại chỉ mà con trỏ b cũng trỏ tới, con trỏ a có giá trị = 3

cout << \*ptr\_a << " " << \*ptr\_b << "\n"; // 3 3

ptr\_c = &ptr\_a; // con trỏ c trỏ đến địa chỉ con trỏ a

cout << \*ptr\_c << " " << \*\*ptr\_c << "\n"; //0xa51510 3

delete ptr\_a;// thu hồi vùng nhớ con trỏ a

ptr\_a = NULL; // con trỏ a = null

system("pause");

}

2. Detect and slove problems of followng program

#include <iostream>

using namespace std;

void main()

{

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

int \*p = a;

int \*p2 = new int;

delete p; // con trỏ p được tạo ra và vẫn ở vùng nhớ stack, nên không thể thu hồi vùng nhớ p vì p chưa được cấp phát bộ nhớ trên heap, chương trình tìm không ra vùng nhớ của p nên sẽ bị treo.

delete a; // mảng a được tạo ra trên stack, nên không được thu hồi vùng nhớ, vì stak không thuộc sự quản lý của lập trình viên.

delete p2;

}

Cách sửa: xóa 2 dòng lệnh delete p; delete a; hoặc cấp phát vùng nhớ cho con trỏ p (int \*p = new int;) và xóa delete a;

3. Using pointer

* Why should we use delete?

Bởi vì khi cấp phát bộ nhớ cho con trỏ trên heap, sử dụng delete để giải phóng vùng nhớ cho con trỏ đó. Khi sử dụng delete con trỏ vẫn còn trên stack nhưng trỏ đến 1 vị trí nào không xác định được trên heap.

* When we use delete?

Khi chúng ta không muốn sử dụng vùng nhớ mà chương trình đã cấp phát trên heap, thì chúng ta trả lại cho chương trình.

* Difference between delete and delete [] ?

Delete là giải phóng vùng nhớ cho con trỏ p, delete [] là giả phóng vùng nhớ cho mảng.

Demo:

int \*p;

p = new int;

delete p; // giải phóng bộ nhớ con trỏ p.

int \*q;

q = new int [10]

delete []q; // giải phóng bộ nhớ cho mảng q, nếu delete q thì chỉ giải phóng cho phần tử đầu của mảng q.

4. Given bellow code

1: \*p1 = 10

2: \*p2 = 0x100

3: \*(\*p2) = 10