**I, POINTER**

1.

#include <iostream>

using namespace std;

typedef int \*IntPtrType;

void main()

{

IntPtrType ptr\_a, ptr\_b, \*ptr\_c;

ptr\_a = new int;

\*ptr\_a = 3; // Gán giá trị vùng nhớ mà ptr\_a trỏ tới = 3

ptr\_b = ptr\_a; // Con trỏ ptr\_b cũng trỏ đến nơi ptr\_a trỏ

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

ptr\_b = new int; // Cấp phát bộ nhớ cho con trỏ ptr\_b

\*ptr\_b = 9; // Gán giá trị vùng nhớ = 9

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

\*ptr\_b = \*ptr\_a; // Gán giá trị vùng nhớ của con trỏ ptr\_b trỏ tới bằng giá trị vùng // nhớ của con trỏ ptr\_a trỏ tới

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

delete ptr\_a; // Giải phóng bộ nhớ mà ptr\_a giữ

ptr\_a = ptr\_b; // Con trỏ ptr\_a cũng trỏ đến nơi ptr\_b trỏ

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

ptr\_c = &ptr\_a; // ptr\_c là con trỏ trỏ đến con trỏ ptr\_a

cout << \*ptr\_c << " " << \*\*ptr\_c << "\n"; //\*ptr\_c : địa chỉ của ptr\_a, \*\*ptr\_c : giá trị vùng nhớ mà ptr\_a trỏ tới = 3 .

delete ptr\_a; // Giải phóng bộ nhớ mà ptr\_a giữ

ptr\_a = NULL; // con trỏ Null

system("pause");

}

2.

Toán tử delete dùng để giải phóng bộ nhớ của một biến đã được cấp phát động. Mảng a được cấp phát bộ nhớ từ Stack nên không thể dùng delete.

Giải quyết:

C1: Không dùng delete p; và delete a;

C2: Cấp phát động cho mảng a: int \*a =new int[4] { 1, 2, 3, 4 };

Để giải phóng bộ nhớ chỉ sử dụng delete [] a; hoặc chỉ sử dụng delete p;

3.

- Sử dụng delete để thu hồi bộ nhớ đã cấp phát tránh lãng phí bộ nhớ.

- Tại bất kỳ thời điểm, khi bạn cảm thấy một biến đã được cấp phát động là không cần thiết nữa, bạn có thể giải phóng bộ nhớ mà nó đã chiếm giữ trong phần bộ nhớ.

- delete được sử dụng để giải phóng bộ nhớ được cấp phát cho một đối tượng.

- delete[] được sử dụng để giải phóng bộ nhớ đã cấp phát cho một mảng các đối tượng.

#include <iostream>

using namespace std;

void main()

{

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

int \*p = new int;

delete[] a;

delete p;

system("pause");

}

4.

\*p1 == 10

\*p2==0x200

\*(\*p2)==10