# MSSV: 18120468

# Họ tên: Lê Hoài Nam

**Bài 1:**

Chương trình chạy bình thường .

Output chương trình:

2

Xuất ra địa chỉ của i

2

Xuất ra địa chỉ của j

Xuất ra địa chỉ của i

**Bài 2:**

Lỗi dòng code thứ 3

Biến con trỏ chỉ nhận địa chỉ, không nhận giá trị

**Bài 3:**

Lỗi dòng 4 và 6. Lỗi biến con trỏ không nhận trực tiếp giá trị

Dòng 8 dòng 9 lỗi. Lý do do lỗi dòng 4 và 6

Dòng 10 lỗi do p2 chưa nhận địa chỉ của một biến nên \*p2 chưa có giá trị nên không thể gán được

Lỗi dòng 14 do p2 chưa nhận địa chỉ của một biến nên \*p2 chưa có giá trị nên không thể in ra được

Lỗi dòng 16 và 18 đều do đều nhận địa chỉ NULL nên không có giá trị để in ra

Output chương trình:

x=1 //output dòng code thứ 7

x=1 //output dòng code thứ 12

p1=1 //output dòng code thứ 13

**Bài 4:**

**Lỗi dòng code thứ 3, biến con trỏ \*p1 chưa nhận địa chỉ nên không thể so sánh với NULL**

**Output chương trình:**

**P2 khác NULL**

**Bài 5:**

Lỗi dòng code thứ 24, vì ch chưa có giá trị nên \*chp nhận địa chỉ của của ch cũng không có giá trị (giá trị rác)

Output chương trình:

1.2 //output dòng code 19

Địa chỉ của z //output dòng code 20

400 //output dòng code 21

Địa chỉ của x //output dòng code 22

100 //output dòng code 23

**Bài 6:**

Lỗi thiếu delete [] a; có thể gây nên rò rỉ dữ liệu, ngoài ra chương trình chạy bình thường .

Output chương trình:

Địa chỉ của a[0] //output dòng code 10

0 //output dòng code 11

Địa chỉ của a[0] //output dòng code 12

50 //output dòng code 13

**Bài 7:**

Lỗi thiếu delete [] p; có thể gây nên rò rỉ dữ liệu, ngoài ra chương trình chạy bình thường .

Không có output

**Bài 8:**

Chương trình chạy bình thường.

Output của chương trình:

0 1 2 3 4 5 6 7 8 9

10 11 12 13 14 15 16 17 18 19

20 21 22 23 24 25 26 27 28 29

30 31 32 33 34 35 36 37 38 39

40 41 42 43 44 45 46 47 48 49

//ouput dòng mã 2 vòng for từ dòng mã 13 đến dòng mã 17

Địa chỉ của mảng p[m] //output của dòng code thứ 5 từ dưới lên

Địa chỉ của mảng p[n] //output của dòng code thứ 4 từ dưới lên

Địa chỉ biến p[1][2] //output của dòng code thứ 3 từ dưới lên

Địa chỉ biến p[1][2] //output của dòng code thứ 2 từ dưới lên

Địa chỉ biến p[5][10] //output của dòng code thứ 2 từ dưới lên

**Bài 9:**

Chương trình chạy bình thường.

Output của chương trình:

0 1 2 3 4 5 6 7 8 9

10 11 12 13 14 15 16 17 18 19

20 21 22 23 24 25 26 27 28 29

30 31 32 33 34 35 36 37 38 39

40 41 42 43 44 45 46 47 48 49

//ouput dòng mã 2 vòng for từ dòng mã 13 đến dòng mã 17

Địa chỉ của mảng p[m] //output của dòng code thứ 4 từ dưới lên

Địa chỉ biến p[2][0] //output của dòng code thứ 3 từ dưới lên

Địa chỉ biến p[2][1] //output của dòng code thứ 2 từ dưới lên

Địa chỉ biến p[3][0] //output của dòng code thứ 2 từ dưới lên

**Bài 10:**

Lỗi dòng code thứ 7, vì \*p1 được khởi tạo là hằng số nên không thể thay đổi giá trị thông qua \*p1, chỉ có thể thay đổi giá trị thông qua x.

Output chương trình:

10 //dòng code thứ 4

100 //dòng code thứ 6

**Bài 11:**

Lỗi dòng code thứ 9, biến con trỏ không thể nhận giá trị.

Lỗi dòng code thứ 11, vì int \* const p khởi tạo biến con trỏ là hằng số không thể thay đổi.

Output chương trình :

10 //output dòng code thứ 4

1000 //output dòng code thứ 6

10000 //output dòng code thứ 8

**Bài 12:**

Chương trình chạy bình thường .

Output chương trình:

5 //ouput dòng code thứ 7

7 //ouput dòng code thứ 9

**Bài 13:**

Chương trình chạy bình thường.

Output chương trình:

5 //ouput dòng code thứ 7

5 //ouput dòng code thứ 9

**Bài 14:**

Chương trình chạy bình thường, trong chương trình không gọi hàm nên giá trị a và b như cũ.

Output chương trình:

a=6.5; b=7.5 //output dòng code thứ 5 trong chương trình chính

**Bài 15:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

typedef struct Sach

{

char Ma[8];

char Ten[51];

char Tacgia[51];

char NXB[51];

float Gia;

int SoTrang;

int NamXB;

};

void NhapSach(Sach \*s)

{

getchar();

printf("Nhap ma sach: ");

gets\_s(s->Ma);

printf("Nhap ten sach: ");

gets\_s(s->Ten);

printf("Nhap ten tac gia: ");

gets\_s(s->Tacgia);

printf("Nhap NXB: ");

gets\_s(s->NXB);

printf("Nhap gia ban: ");

scanf\_s("%f", &s->Gia);

printf("Nhap so trang: ");

scanf\_s("%d", &s->SoTrang);

printf("Nhap nam xuat ban: ");

scanf\_s("%d", &s->NamXB);

}

void Swap(Sach &x,Sach &y)

{

char temp[51];

int temp1;

strcpy\_s(temp, y.Ma);

strcpy\_s(y.Ma, x.Ma);

strcpy\_s(x.Ma, temp);

strcpy\_s(temp, y.Ten);

strcpy\_s(y.Ten, x.Ten);

strcpy\_s(x.Ten, temp);

strcpy\_s(temp, y.Tacgia);

strcpy\_s(y.Tacgia, x.Tacgia);

strcpy\_s(x.Tacgia, temp);

strcpy\_s(temp, y.NXB);

strcpy\_s(y.NXB, x.NXB);

strcpy\_s(x.NXB, temp);

temp1 = y.Gia;

y.Gia= x.Gia;

x.Gia = temp1;

temp1 = y.SoTrang;

y.SoTrang = x.SoTrang;

x.SoTrang = temp1;

temp1 = y.NamXB;

y.NamXB = x.NamXB;

x.NamXB = temp1;

}

void SapXep(Sach \*p,int n)

{

for (int i = 0; i < n; i++)

{

for (int j = i + 1; j < n; j++)

if (p[i].NamXB > p[j].NamXB)

Swap(p[i], p[j]);

}

}

void XuatSach(Sach \*s)

{

printf("Ma sach: %s\n", s->Ma);

printf("Ten sach: %s\n", s->Ten);

printf("Tac gia: %s\n", s->Tacgia);

printf("Nha xuat ban: %s\n", s->NXB);

printf("Gia sach: %.2f\n", s->Gia);

printf("So trang: %d\n", s->SoTrang);

printf("Nam xuat ban: %d\n", s->NamXB);

}

void NhapMangSach(Sach \*p,int n)

{

for (int i = 0;i < n;i++)

{

printf("\nNhap quyen sach %d\n", i + 1);

NhapSach(p);

p = p + 1;

}

}

void XuatMangSach(Sach \*p, int n)

{

printf("\nXuat cac sach: \n");

for (int i = 0;i < n;i++)

{

printf("\nXuat quyen sach thu %d: \n", i + 1);

XuatSach(p);

p = p + 1;

}

}

void TimSach(Sach \*p,int &k,int n)

{

printf("Nhap k: ");

scanf\_s("%d", &k);

Sach\* temp = p;

for (int i = 0;i < n ;i++)

{

if (p->SoTrang > k)

continue;

else

{

if (p->Gia > temp->Gia)

temp = p;

}

}

printf("Quyen sach can tim la: \n");

XuatSach(temp);

}

void main()

{

Sach \*p = NULL;

int n, k, nam;

printf("Nhap vao so luong sach: ");

scanf\_s("%d", &n);

p= new Sach[n];

NhapMangSach(p, n); //cau 15.1

XuatMangSach(p, n); //cau 15.2

SapXep(p, n); //cau 15.3

TimSach(p, k, n);//cau 15.4

//chua lam xong bai 15.5 va 15.6, neu lam bang vevtor se de hon...

delete[] p;

\_getch();

}