MSSV: 1512418

Họ tên: Trần Duy Phương

**Bài 1:**

int i=2;

int \*j;

j = &i;

printf("%d\n", i);// xuất ra giá trị biến i = 2

printf("%d\n", j);// xuất ra địa chỉ của i

printf("%d\n", \*j);//xuất ra giá trị vùng nhớ j trỏ đến = 2

printf("%d\n", &i);//xuất ra địa chỉ biến i

printf("%d\n", &j);//xuất ra địa chỉ của j

**Bài 2:**

int \*a = new int;

int \*b = new int;

\*a = 2;//thay đổi giá trị vùng nhớ a trỏ đến là bằng 2

b = a;//truyền cho b địa chỉ của a

printf("%d\n", \*a);//xuất ra giá trị của vùng nhớ a trỏ đến bằng 2

printf("%d\n", \*b);///xuất ra giá trị của vùng nhớ b trỏ đến bằng 2

delete a;

delete b;

**Bài 3:**

int x, \*p1, \*p2;

x = 1;

p1 = new int;

\*p1 = 5;

p2 = new int;

\*p2 = 3;

printf("X = %d\n", x);// xuất ra giá trị biến x=1

printf("p1 = %d\n", \*p1);//xuất ra giá trị của vùng nhớ p1 trỏ đến bằng 5

printf("p2 = %d\n", \*p2);//xuất ra giá trị của vùng nhớ p2 trỏ đến bằng 3

x = \*p2;

p1 = &x;

printf("X = %d\n", x);// xuất ra x=3 vì nó đã được gán bằng giá trị p2 trỏ đến

printf("p1 = %d\n", \*p1);//xuất ra giá trị của vùng nhớ p1 trỏ đến bằng 1 vì đã truyền cho nó địa chỉ của x=3

printf("p2 = %d\n", \*p2); //vẫn không đổi bằng 3

p1 = NULL;//con trỏ không trỏ đến vùng nhớ nào nên xuất ra sẽ bị lỗi

printf("p1 = %d\n", \*p1);

p2 = 0;// tương tự như p2 = NULL

printf("p1 = %d\n", \*p2);

delete p1;

delete p2;

**Bài 4:**

int \*p1;//chưa khởi tạo vùng nhớ

int \*p2 = new int;

if (p1 = NULL)

printf("p1 bang NULL");

else

printf("p1 khac NULL");

//sẽ xuất ra p1 khác NULL

if (p2 = NULL)

printf("p2 bang NULL");

else

printf("p2 khac NULL");

//sẽ xuất ra p2 khác NULL

**Bài 5:**

int x, z;  
float y;  
char ch, \*chp;  
int \*ip1, \*ip2;  
float \*fp;  
x = 100;  
y = 20.0;  
z = 50;  
ip1 = &x;// truyền địa chỉ của x cho ip1  
ip2 = &z;// truyền địa chỉ của z cho ip2  
fp = &y;// truyền địa chỉ của y cho fp  
chp = &ch;// truyền dịa chỉ của ch cho chp  
ip2 = ip1;//địa chỉ ip2 lúc này bằng ip1, \*ip2 = 100  
ip1 = &z;// truyền địa chỉ của z cho ip1, \*ip1 = 50  
\*ip1 = \*ip2;// \*ip1 =100  
\*ip1 = 200;//\*ip1 = 200  
\*ip1 = \*ip2 + 300;// \*ip1 =400  
\*fp = 1.2;// \*fp = 1.2  
printf("%f\n", \*fp);// \*fp = 1.2  
printf("%d\n", ip1);// địa chỉ ip1  
printf("%d\n", \*ip1);// giá trị vùng nhớ ip1 trỏ tới là \*ip1 = 400  
printf("%d\n", ip2);//địa chỉ ip2  
printf("%d\n", \*ip2);// giá trị vùng nhớ ip2 trỏ tới là \*ip1 = 100  
printf("%c\n", \*chp);// // giá trị vùng nhớ chp trỏ tới

**//bai 6**  
int \*a;  
int x;  
printf("x = ");  
scanf("%d", &x);  
a = new int[x];  
for (int i = 0; i < x; i++)  
{  
a[i] = i \* 10;  
}  
printf("a=%d\n", a);// a = địa chỉ phần tử đầu mảng  
printf("a=%d\n", \*a);// \*a = 0 là giá trị của phần tử đầu mảng  
printf("a+1=%d\n", a);// vẫn là địa chỉ phần tử đầu mảng  
printf("a+5=%d\n", \*(a+5));// \*(a+5) = 50 giá trị phần tử thứ 5 của mảng

**//bai7**

int \*p;  
for (int i = 0; i < 10000; i++)  
{  
p = new int[100000];// cấp phát bộ nhớ là 100000  
}

**//bai 8**  
 int \*\*p;

int m, n;

scanf("%d", &m);

scanf("%d", &n);

p = new int\*[m];

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

p[i] = new int[n];

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

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

p[i][j] = i\*n + j;

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

{

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

printf("%4d", p[i][j]);

printf("\n");

}

//giả sử nhập m = 5, n = 10

printf("%d\n", p);//p xuất ra không là địa chỉ phần tử đầu của mảng

printf("%d\n", \*p);// \*\*p mới là giá trị của phần tử đầu của mảng = 0

printf("%d\n", \*(p + 2));

//\*\*(p + 2) la gia tri cua phần tử thứ 2 nên \*(p + 2) sẽ xuất ra địa chỉ rác

printf("%d\n", \*(p + 10));

// \*\*(p + 10) la gia tri cua phần tử thứ 10 nên \*(p + 10) sẽ xuất ra địa chỉ rác

printf("%d\n", \*(p + 49));

//\*\*(p + 49) la gia tri cua phần tử thứ 49 nên \*(p + 49) sẽ xuất ra địa chỉ rác

**//bai 9**

int \*\*p;

int m, n;

scanf("%d", &m);

scanf("%d", &n);

p = new int\*[m];

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

p[i] = new int[n];

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

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

p[i][j] = i\*n + j;

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

{

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

//xuống dòng 50 lần

printf("\n");

}

//giả sử nhập m = 5, n = 10

printf("%d\n", \*\*p);// \*\*p là giá trị của phần tử đầu của mảng = 0

printf("%d\n", \*\*(p + 2));//\*\*(p + 2) la gia tri cua phần tử thứ 2 \*\*p = 20

printf("%d\n", \*(\*(p + 2)+1));// \*\*(p + 10) la gia tri cua phần tử thứ 10 nên \*(p + 10) sẽ xuất ra địa chỉ rác

printf("%d\n", \*(p + 3));//\*\*(p + 3) la gia tri cua phần tử thứ 3 nên \*(p + 3) sẽ xuất ra địa chỉ rác

**//bài 10:**

int x = 10;

const int\* p1;

p1 = &x;

printf("%d\n", \*p1);// xuất ra p1 = 10

x = 100;

printf("%d\n", \*p1);// xuất ra p1 = 100

\*p1 = 1000;// lỗi vì con trỏ \*p1 là hằng số nên không thay đổi giá trị được

printf("%d\n", \*p1);

**//Bài 11:**

int x = 10;

int y = 5;

int\* const p = &x;

printf("%d\n", \*p);//xuất ra \*p = 10 là giá trị vùng nhớ mà con trỏ p trỏ đến

x = 1000;

printf("%d\n", \*p);

//xuất ra \*p = 1000 là giá trị vùng nhớ mà con trỏ p trỏ đến khi x thay đổi p thay đổi theo và bằng 1000

\*p = 10000;

printf("%d\n", \*p);

//xuất ra \*p = 10000 là giá trị vùng nhớ mà con trỏ p trỏ đến khi thay đổi \*p bằng 10000

p = 7;// lỗi vì truyền địa chỉ cho con trỏ p không hợp lệ

printf("%d\n", \*p);

p = &y;// bị lỗi vì p lúc đầu khai báo là int\* const rồi nên không thay đổi được

printf("%d\n", \*p);

**//bai12:**

void f1(int \*p, int n)

{

\*(p + 5) = 7;

}

void main()

{

int \*p;

int n;

scanf("%d", &n);

//giả sử nhập n = 10

p = new int[n];

\*(p + 5) = 5;

printf("%d\n", \*(p + 5));// xuất ra \*(p+5) = 5

f1(p, n);

printf("%d\n", \*(p + 5));

// xuất ra \*(p+5) = 7 vì hàm f1 truyền vào 1 con trỏ và không có cấp phát nên nó sẽ thay đổi khi gọi hàm

}

**//bai 13**

void f2(int \*p, int n)

{

p = new int[12];

\*(p + 5) = 7;

}

void main()

{

int \*p;

int n;

scanf("%d", &n);

//giả sử nhập n = 10

p = new int[n];

\*(p + 5) = 5;

printf("%d\n", \*(p + 5));// xuất ra \*(p+5) = 5

f2(p, n);

printf("%d\n", \*(p + 5));// xuất ra \*(p+5) = 5 vì chưa truyền tham chiếu cho hàm

}

**//bai 14**

void hoanvi(float \*x, float \*y)

{

float t;

t = \*x;

\*x = \*y;

\*y = t;

}

float a, b;

a = 6.5;

b = 7.5;

hoanvi(&a, &b);//khi truyền địa chỉ vào hàm thì mới xuất ra kết quả a = 7.5, b =6.5

printf("a=%f; b=%f\n", a, b);//bị lỗi nên phải thay %d trong đề thành %f để xuất ra số thực