ID: 25125083

Student name: Nguyễn Đình Minh Huy

Class: 25A01

In-class assignments

Week: 07

Assignment 1:

```
int a = 3;
int *b = &a; // assign the address of a to pointer b
cout << b << endl; // Ouput: the address of stored in b (address of a)
cout << *b << endl; // Output: the value at the address at b (value of a)
cout << &b << endl; // Output: the address of pointer b
cout << a << endl; // Output: value of a
cout << &a << endl; // Output: address of a
```

Suppose address of a: 0x100, address of b: 0x200, then the **total output** is:

```
0x100
3
0x200
3
0x100
```

There are no errors.

Assignment 2:

Suppose: Addresses of variables: &x: 0x100, &y: 0x200, &z: 0x300, &ch: 0x400.

Addresses of pointers: &ip1: 1x100, &fp: 1x200, &ip2: 1x300, &chp: 1x400

```
int x,z;
float y;
char ch, *chp;
int *ip1, *ip2;
float *fp;

x = 100;
y = 20.0;
z = 50;
ch = 'Z';
// set value of variables

ip1 = &x;
ip2 = &z;
fp = &y;
chp = &ch;
//  set address stored in pointers to variables

ip2 = ip1; // ip2 = &x
ip1 = &z; // ip1 = &z
*ip1 = *ip2; // set value: z = x (=100)

*ip1 = 200; // z = 200
*ip1 = *ip2 + 300; // z = x + 300 = 100 + 300 = 400
*fp = 1.2; // y = 1.2
```

```
cout << x << endl; // Output: 100
cout << y << endl; // Output: 1.2 (with formatted precision)
cout << z << endl; // Output: 400

cout << ip1 << endl; // Output: 0x300
cout << *ip1 << endl; // Output: 400
cout << &ip1 << endl; // Output: 1x100

cout << ip2 << endl; // Output: 0x100
cout << *ip2 << endl; // Output: 100
cout << &ip2 << endl; // Output: 1x300

cout << fp << endl; // Output: 0x200
cout << *fp << endl; // Output: 1.2 (with formatted precision)
cout << &fp << endl; // Output: 1x200

cout << chp << endl; // Output: 0x400
cout << *chp << endl; // Output: Z
cout << &chp << endl; // Output: 1x400
```

Assignment 3:

```
int *a = new int; // allocate memory

int *b = new int; // allocate memory

*a = 2; // set value at the address stored in a

b = a; // set the address stored at b to the address at a (a and b point to the same address)

cout << *a << endl; // Output: 2

cout << *b << endl; // Output: 2

delete a; // Deallocate memory

delete b; // Error: try to free memory the second time
```

Note: Memory leak at the initial address stored in b

The **total output** is:

```
2

2
```

There are errors above.

Assignment 4:

```
int a = 3;

int *p = &a; // set the address stored in p to address of a

cout << *p << endl; // Output: 3 (value at the address stored in p (value of a))

p = new int(5); // Allocate and set value at new memory to 5

cout << *p << endl; // Output: 5 (value at the address stored in p (new memory))
```

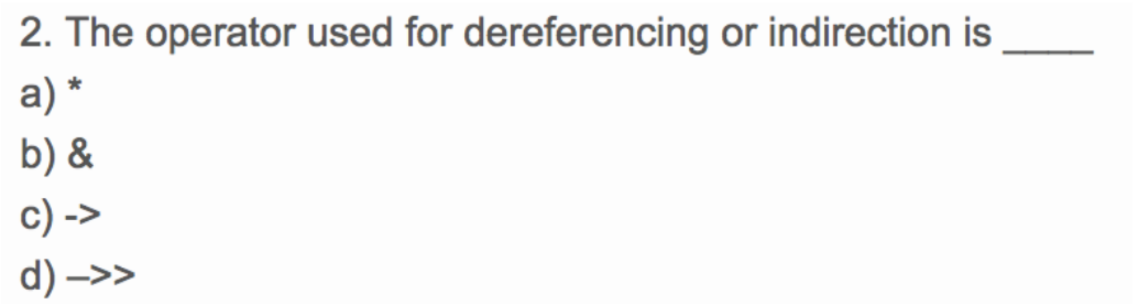
The **total output** is:

3

5

There are no errors.

Assignment 13:



2. The operator used for dereferencing or indirection is ____

a) *

b) &

c) ->

d) –>>

Answer: a

***** is dereferencing operator which is used to extract the value at the address stored in the pointer