

1.) What is the output?

```
int x = 20;
int* p = &x;
cout << p << endl;
cout << *p << endl;
```

address of x
20
or p = address of x
don't know, actually it's come from computer memory address

2.) What is the output?

```
int x = 60;
int* p = &x;
x = x + 5;
*p = x + 5;
*p = *p + 5;
cout << *p << endl;
cout << "The address of p is " << &*p << endl;
```

Alias = AKA *alias*
ex. int num = 10;
int x;
int *p;
p = #
x = *p;

75
The address of p is ~~75~~ don't know, as I mention above
The address of p is p

3.) Assuming pointer p1 and p2 is created as integer variable type. What is the different between p2 = p1 and *p2 = *p1.

int *p1;
int *p2;
p2 = p1
: is the address
Ex. pointer p2 is pointing at the same address as p1
*p2 = *p1
: variable of pointer p2 will be equal to variable of pointer p1
assign value of pointer p1 to p2

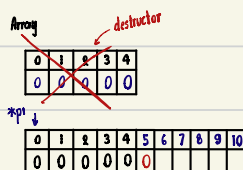
capacity : index ที่ขนาดที่ vector มี

4.) What is '&' and is it different from '*'. Why?

& : is reference to ~~address~~
* : is pointer sign use to indicate that it's pointer ~~content~~
L use to refer to something in that content (use in pointer)
L variable

<name>.capacity
reserve : reserve capacity
ex. reserve.size() - 10 จำนวน 10 ที่
resize : เพิ่มจำนวนที่เก็บข้อมูล

int *p; is same as int* p;



array int x[10] = {1};
หมายถึง ใช้ pointer ที่ index ที่ 0 แล้วคูณ 2 → 10 x 2 = 20
ขนาด pointer 10 บิต

Vector
0 1 2 3 ← x2 ไม่เต็ม
0 0 0 0