

Phạm Văn Đức

7.1 - PO1.

#include &lt;stdio.h&gt;

```
int main() {
    int *x = new int;
    int y = 2;
    int *z = &y;
    *x = *z + y;
    cout << *x << endl;
    delete x;
    return 0;
}
```

Giá trị của  $x$  là  $y + z$ , tức là  $2 + 2 = 4$   
trong bảng trỏ của  $x$  là công y là 4

## 7.2 PO2

Hàm swap<sup>1</sup> sử dụng pass by value, giá trị của biến được sao chép. Giá trị này thay đổi không ảnh hưởng đến ngoài hàm.

Hàm swap<sup>2</sup> sử dụng pass by reference, tham chiếu tới địa chỉ của biến, gọi là nó sẽ thay đổi các biến gốc.

Hàm swap<sup>3</sup> sử dụng pass by pointer, nhận con trỏ đến các biến cần đổi, theo đó thay đổi làm thay đổi các biến gốc.

## 7.3. PO3.

P1 trỏ đến vị trí phần tử m[0]

P2 trỏ đến vị trí phần tử m[6]

P2 - P1 + 6 → khoảng cách giữa 2 phần tử theo m là 6 phần tử double. Tức San khi truy cập thì sẽ chỉ cho đối tượng 1 phần tử double.

fig chỉ biến sẽ ép thành kiểu trả char,  
 và đưa vào trả p. Cách hoạt động như 1 xâm.  
 Vì char chỉ có dung lượng 1 byte và ta cho  
 in theo dạng số nguyên ở printf nên  
 ta sẽ in ra từng giá trị byte của số  
 ở dạng số nguyên

Theo cách trái little-endian : -1300

Theo cách trái big-endian ; 003-1.

~~Đã~~

Yêu Cầu

Kết quả in ra là 120

vì sizeof trả kích thước theo byte, để  
 tăng tới độ sâu lít thì cái padding được  
 thêm vào đี đầu bởi số 0 và 8.