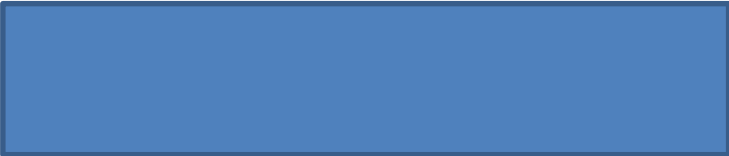


## HW1 포인터 사용하여 문자열 길이 구하기

```
#include <iostream>
#include <string>

using namespace std;
int len(char* s){

}

int main( ){
    char string[100];
    cout<<"enter:";
    gets(string);
    cout<<len(string)<<endl;
}
```

## HW2 포인터로 함수 매개변수 넘기기

### 요구사항

- ① 다음과 같이 학생 10명의 수학 성적이 있다.
  - a. 학생 번호: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
  - b. 수학 성적: 90, 88, 85, 55, 47, 92, 87, 30, 89, 55
- ② 수학 성적이 가장 높은 학생을 찾아라.
- ③ 수학 성적은 정수로 한정한다.
- ④ 정수형 배열을 사용하라.

```
int FindMaxNumber(int* argGrade);
```

```
void main() {  
    int grade[10] = { 90, 88, 85, 55, 47, 92, 87, 30, 89, 55 };  
    int maxGradeIndex = 0;  
    maxGradeIndex = FindMaxNumber(grade);  
    cout << "Best grade = " << grade[maxGradeIndex] << '\n';  
    cout << "StudentID = " << maxGradeIndex + 1 << '\n';  
}
```

## HW3 2개 이상의 값을 리턴 받을 수 있을까?

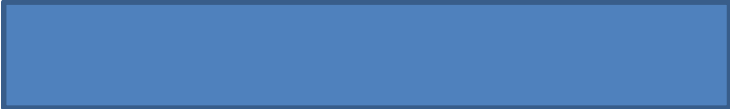
```
#include <iostream>

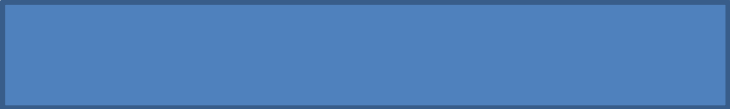
using namespace std;

void foo(int x, int y, int *x1, int *x2);
void foo2(int x, int y, int *x1, int *x2);

int main(){
    // compute quotient and remainder
    int q;
    int r;
    foo(23, 7, &q, &r);
    cout<<"23/7: quotient:"<<q<<"remainder:"<<r<<endl;

    // compute sum and difference
    int sum, diff;
    foo2(23, 7, &sum, &diff);
    cout<<"23+7="<<sum<<" 23-7="<<diff<<endl;
}

void foo(int x, int y, int *x1, int *x2){
    // divide x by y and put quotient in *x1, remainder in *x2
    
}

void foo2(int x, int y, int *x1, int *x2){
    // the result of x+y will be returned in *x1, the result of x-y in *x2
    
}
```

## HW4 메모리 변화 그림으로 그리기

```
#include <iostream>

using namespace std;

void swap1 (int left, int right)
{
    int temp = left;
    left = right;
    right = temp;
}

void swap2 (int *p_left, int *p_right)
{
    int temp = *p_left;
    *p_left = *p_right;
    *p_right = temp;
}

int main ()
{
    int x = 1, y = 2;
    swap1( x, y );
    cout << x << " " << y << '\n';
    swap2( & x, & y );
    cout << x << " " << y << '\n';
}
```

swap1과 swap2에서  
변수 및 포인터의 메모리 저장  
모습, 함수의 시작 및 끝에 따라  
바뀌는 모습 등을 그림으로 그  
리세요.