

\*\*\*\*\* 결과 출력은 출력 예시를 따릅니다. \*\*\*\*\*  
4번 5번은 주어진 스켈레톤 코드에 최대한 맞추어 작성하시기 바랍니다.

[1] 주어진 문자열에 대해 대소문자 변환 2회 실행하여 각각의 결과를 출력하십시오.

1. 문자열 : "Ice Vanilla Latte"를 배열에 저장합니다.
2. 문자열의 대문자를 소문자로, 소문자를 대문자로 변환하여 출력
3. 변환된 문자열의 대문자를 소문자로, 소문자를 대문자로 변환하여 출력
4. 대문자 -> 소문자 변환은 32를 더해줌
5. 소문자 -> 대문자 변환은 32를 빼줌
6. 문자열의 최대 길이는 100 입니다.

문제 출력 예시

```
original text : [ice Vanilla Latte]
ICE vANILLA LATTE
ice Vanilla Latte
```

## [2] 입력받은 문자열에서 중복되는 문자를 찾기

1. 사용자로부터 2개의 문자열을 입력받음
2. 입력받은 2개의 문자열에서 같은 문자를 찾아 출력
3. 대소문자를 구분함
4. 문자열의 최대 길이는 100 입니다.

입력 예시)

문자열 1 :

abvnzx,mvbnxcvbn,mxz**p**bnv!@#\$^&\*@!@#^@&#zmxcbvcnmxz**p**bnz-=-  
vbnm[x**l**]b]c[n[[[]]][[v]x[]mb]zm**e**

문자열 2 :

tywqiw5t6y8qi9w9t9qy7i8**a**5w1t2y3q5w6w6t8q8y8u8w8475878419647138tqyi**p**tqy  
**p**wt4q5y54i6**l**q6w6ty8iy9u9w9q9t7y8u2i1**e**

결과 출력 예시

**apple**

### [3] 성적 관리 프로그램

1. “국어”, “영어”, “수학”, “과학” 4개의 과목에 대한 성적 관리 프로그램 작성
2. 4명의 학생 이름을 입력받아 ~~name~~ 배열에 저장
3. 4명 학생의 4과목 점수 (1~9)를 입력받아 score 배열에 저장
4. 각 학생의 총점과 과목별 총점을 계산하여 저장하고, 출력

(1) 입력 받은 점수와 총점은 5X5 2차원 score 배열에 저장하여 출력

(2) main()을 포함하여 총 5개의 함수로 구성되는 프로그램으로 구현

- A. 학생의 이름을 저장할 배열을 전달받아 학생의 이름을 입력받고, 배열에 저장하는 input\_name() 작성
- B. 학생의 이름과 배열을 전달받아 학생의 점수를 입력받고, 배열에 저장하는 input\_score() 작성. input\_score()는 추가적인 매개변수를 사용할 수 있다.
- C. 학생들의 점수가 저장된 배열을 전달받아 학생 별, 과목 별 총점을 계산하고, 배열에 저장하는 calc\_total()를 작성
- D. 학생들의 점수와 총점이 저장된 배열을 전달받아 각 학생 별, 과목 별 총점을 출력하는 print\_score()를 작성
- E. main()은 배열을 생성하고, 필요한 함수들을 호출하여 프로그램을 완성

4번 5번은 주어진 스켈레톤 코드에 최대한 맞추어 작성하시기 바랍니다.

[4] 입력 받은 문자를 정렬하는 프로그램[주어진 스켈레톤 코드를 사용하여 구현하시오.]

1. 사용자로부터 최대 100개의 문자와 정렬 방식을 입력받고, 정렬 결과를 출력하는 프로그램 작성
  - A. 사용자로부터 문자들과 정렬 방식을 입력받는 input\_char()를 작성
  - B. 사용자는 정렬 방식을 "오름차순", "내림차순"으로 입력
  - C. 매개변수로 받은 문자열과 정렬 방식을 바탕으로 문자들을 정렬하고 배열에 저장하는 func\_sort() 작성
  - D. 정렬된 배열을 전달받아, 배열의 숫자를 출력하는 func\_print() 작성
  - E. main()은 필요한 배열을 생성하고, 함수들을 호출하여 프로그램을 완성

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

int ARR_MAX_SIZE = 100;

char input_char(char *target_string, char
*sorting_type);
void func_sort(char *target_string, char
*sorting_type);
void func_print(char *target_string);

int main(){
    char target_string[ARR_MAX_SIZE];
    char sorting_type[ARR_MAX_SIZE];
    /*write your code here*/
    input_char(/*write your code here*/);
    func_sort(/*write your code here*/);
    func_print(/*write your code here*/);
}

char input_char(target_string, sorting_type){
    /*write your code here*/
}

void func_sort(target_string, sorting_type){
    /*write your code here*/
}

void func_print(target_string){
    /*write your code here*/
}
```

**[5]** xy 평면의 좌표를 입력받고 x축 대칭, y축 대칭, 원점대칭을 순서대로 진행하여 출력하는 프로그램을 구현하십시오. **[주어진 스켈레톤 코드를 사용하여 구현하십시오.]**

4X4 16칸으로 구성된 좌표 평면의 좌표는 다음의 이미지와 같다

				2
				1
				-1
				-2
-2	-1	1	2	

프로그램의 구동 순서는 다음과 같다.

- (1) 좌표 평면위의 한 점의 x, y 좌표를 입력 받는다.

```
put X coordination : 1
put Y coordination : 2
```

- (2) 입력 받은 좌표는 'X'로, 그렇지 않은 좌표는 'O'으로 4X4 좌표 평면을 출력한다.

```
OOXO
OOOO
OOOO
OOOO
```

- (3) 입력 받은 좌표를 x축으로 대칭 변환하고, 4X4 좌표 평면을 출력한다.

```
OOOO
OOOO
OOOO
OOXO
```

- (4) 변환된 좌표를 다시 y축으로 대칭 변환하고, 4X4 좌표 평면을 출력한다.

```
OOOO
OOOO
OOOO
OXOO
```

- (5) 변환된 좌표를 다시 원점으로 대칭 변환(x축, y축 대칭 변환)하고, 4X4 좌표 평면을 출력한다.

```
OOXO
OOOO
OOOO
OOOO
```

또다른 출력 예시는 다음과 같다.

```
put X coordination : -1
put Y coordination : -2
```

```
OOOO
OOOO
OOOO
OXOO
```

```
OXOO
OOOO
OOOO
OOOO
```

```
OOXO
OOOO
OOOO
OOOO
```

```
OOOO
OOOO
OOOO
OXOO
```

스켈레톤 코드

```
#include <stdio.h>
```

```
// 사용자로부터 x, y 좌표값을 입력 받는 함수
```

```
void get_coordination(int *x_point, int *y_point);
```

```
// 전달받은 좌표를 x축으로 대칭 변환하는 함수
```

```
void change_x_axis(int *x_point, int *y_point);
```

```
// 전달받은 좌표를 y축으로 대칭 변환하는 함수
```

```
void change_y_axis(int *x_point, int *y_point);
```

```
// 전달받은 좌표를 원점으로 대칭 변환하는 함수
```

```
void change_xy_axis(int *x_point, int *y_point);
```

```
// 4X4 좌표 평면을 출력하는 함수
```

```
void print_matrix(int x_point, int y_point);
```

```
// 전달받은 좌표에 대해 (2)~(5)을 수행하는 함수
```

```
void show_matrix(int x_point, int y_point);
```

```
int main(){

    int x_point, y_point;

    get_coordination(&x_point, &y_point);
    show_matrix(x_point, y_point);

    return 0;
}

void get_coordination(int *x_point, int *y_point){
    /*write your code here*/
}

void change_x_axis(int *x_point, int *y_point){
    /*write your code here*/
}

void change_y_axis(int *x_point, int *y_point){
    /*write your code here*/
}

void change_xy_axis(int *x_point, int *y_point){
    /*write your code here*/
}

void print_matrix(int x_point, int y_point){
    /*write your code here*/
}

void show_matrix(int x_point, int y_point){
    /*write your code here*/
}
```