

■ 서버 제출 코드

유의사항을 잘 읽고 제출합니다. 제출하는 코드는 다음 세가지 중의 하나입니다.

1. 함수 A만 제출하라고 한 경우 함수 A만 서버 창에 제출
2. 코드 전체 제출
3. main함수와 선언(include 문장, define 문장, 타입 정의 포함)을 *제외*한 사용자 정의 함수들에 대한 코드 제출

■ 문제, 난이도 및 배점

재귀	(난이도 하)	문제1	5
문자문자열(포인터)	(난이도 하)	문제2_1	5
	(난이도 중)	문제2_2	5
문자열	(난이도 중상)	문제3	15
구조체	(난이도 하)	문제4_1	5
	(난이도 중상)	문제4_2	10
	(난이도 중상)	문제4_3	10
오목	(난이도 중상)	문제5	15

/* 문제 1: 각 자리수의 합

정수를 입력 받아 각 자리수의 합을 구하여 출력하는 프로그램을 위해
addEach 함수를 *재귀적*으로 작성하라.

실행예 #1

```
123 <- 입력
```

```
6 <- 출력
```

실행예 #2

```
1234 <- 입력
```

```
10 <- 출력
```

유의사항

- addEach 함수만 제출한다

- 채점시 코드 체크함

1. 재귀로 작성하지 않고 제출할 시 0점 처리함
2. 재귀함수안에서 반복문을 사용하거나 static 함수, 전역변수를 사용할 시 0점 처리함

```
*/
```

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

```
int addEach(int n) // 이 함수를 작성해서 제출
```

```
{
```

```
    // 작성
```

```
}
```

```
int main(void) // 변경하지 말라
```

```
{
```

```
    int num;
```

```
    scanf("%d", &num);
```

```
    printf("%d\n", addEach(num));
```

```
}
/*
```

문제 2-1: 문자문자열 1

문자열을 읽어서 이를 정수로 변환하여 출력하는 프로그램을 작성하려 한다.

이때, 문자열에 하나라도 숫자 아닌 것이 섞여있으면 0으로 변환하여 출력한다.

문자열이 모두 숫자로 되어있으면 그것을 정수로 변환한 값을 반환하고

문자열에 하나라도 숫자 아닌 것이 섞여 있으면 0을 반환하는 함수 strToNum을 정의하라.

실행예1

```
1234 <- 문자열로 입력
```

```
1234 <- 정수로 출력
```

실행예2

```
123a <- 문자열로 입력
```

```
0 <- 문자a가 섞여있으므로 0을 출력
```

실행예3

```
aaaa <- 문자열로 입력
```

```
0 <- 출력
```

유의사항

- **strToNum만 제출한다.**

- 원하면 문자처리 함수를 사용한다.

- **채점시 코드 체크함**

코드를 살펴보고 아래의 사항에서 위반되면 0점 처리합니다.

1. 문자열함수는 사용하지 않는다.

2. 아래와 같은 표현은 사용하지 않는다.

= 배열적 표현: 즉 `s[i]`같은 형태는 사용하지 않는다.

= `*(s + i)`같은 표현도 사용하지 않는다. 즉, `s`값을 직접 변화하면서 정의한다.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
int strToNum(char *s) // 이 함수만 제출
{
    // 작성
}
int main(void) // main은 제공
{
    char str[50];

    scanf("%s", str);
    printf("%d\n", strToNum(str));
}
```

/*

문제 2_2: 문자문자열2

sentence(공백 문자 포함)를 입력 받아

제시된 구분자를 기준으로 단어(토큰, 문자열)로 분리한 후 숫자로만 구성된 문자열들의 합을 구하여 출력하라.

실행예

111 11aa 222 bbb <---- 입력: 하나의 문장으로 읽는다. (gets사용)

333 <---- 출력: 111 + 222

유의사항

- 코드 전체를 제출한다

- 앞에서 작성한 strToNum을 사용해도 된다(물론 안해도 된다).
- 자유로이 프로그래밍한다. 즉 문자함수, 문자열 함수를 어떤 것이든 사용할 수 있다.

*/

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <ctype.h>
```

```
int main(void)
```

```
{
```

```
    char sentence[50]; // 총 49이하의 문자들로 문장 구성
```

```
    char delimiters[] = " ,\n\t0"; // 네 개의 분리 문자들(공백, 콤마, 줄바꿈, 널문자) 지정한다
```

```
    char *token;
```

```
    int sum = 0;
```

```
    gets(sentence);
```

```
    // 이하 코드 작성
```

```
    printf("%d\n", sum);
```

```
}
```

/*

문제 3: 문자열 나누기

입력 받은 문자열 str에서
앞에서 2개씩 문자들을 추출해서
다음과 같이 두개의 문자열 s1, s2를 만들려 한다.

입력이 aabbccddeeffgg인 경우
s1은 aacceegg
s2는 bbddff
로 만들어진다. 마지막에 문자가 1개 남는 경우도 그대로 해당 문자열에 추가한다/

실행예 #1

```
aabbccddeeffgg <- 입력: 문자열
aacceegg <- 출력: s1
bbddff <- 출력: s2
```

실행예 #2

```
abcdefghijkl <- 입력: 문자열
abefij <- 출력: s1
cdghk <- 출력: s2
```

유의사항

- splitString 함수만 제출한다

*/

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

```
void splitString(char *str, char *s1, char *s2) // 이 함수만 제출
{
    // 작성
}
```

```
int main(void)
{
    char str[50], s1[50], s2[50];

    scanf("%s", str);
    splitString(str, s1, s2);

    printf("%s\n", s1);
    printf("%s\n", s2);
}
```

/*

문제 4_1: 구조체 1

학생 정보는 이름, 중간고사, 학기말고사, 학점으로 되어있다.

프로그램은 다음과 같다

1. 새로운 학생의 정보(이름, 중간고사, 학기말고사)를 반복적으로 (1은 새 회원 정보 계속 입력, -1은 끝을 의미) 읽어 들인 후
2. 모든 학생에 대해서 학점을 계산한다.
중간고사와 학기말고사 합이 100 초과인 경우 A, 아니면 F이다.
3. 현재까지의 학생들의 정보를 출력한다.

실행예 #1

```
1          <--- 새 정보 입력
ccc 30 90
1          <--- 새 정보 입력
DDD 50 50
1          <--- 새 정보 입력
fff 100 100
-1         <--- 정보 입력 끝
ccc 30 90 A <--- 이하 출력: 공백을 한 개씩 정확하게 넣어서 출력한다
DDD 50 50 F
fff 100 100 A
```

실행예 #2

```
1          <--- 새 정보 입력
AAA 30 70
1          <--- 새 정보 입력
DDD 30 71
1          <--- 새 정보 입력
fff 100 50
1          <--- 새 정보 입력
zzz 100 51
-1         <--- 정보 입력 끝
AAA 30 70 F <--- 이하 출력
DDD 30 71 A
fff 100 50 A
zzz 100 51 A
```

유의사항

- main 함수와 선언(include 문장, define 문장, 타입 정의 포함)을 *제외*한 사용자 정의 함수들에 대한 코드를 제출한다.

- 학생 정보를 출력하는 printStudent에서 이름, 중간고사, 학기말, 학점을 출력할 때 1개의 공백을 정확히 사용한다.

즉 printf에서 다음과 같이 출력되게 한다.

이름(공백)중간(공백)학기말(공백)학점(줄바꿈) */

```
#include <stdio.h>
#define MAX 20

typedef struct {
    char name[MAX];
    int mid;
    int final;
    char grade;
} Student;

void printStudent(Student *m) // 작성해서 제출
{
    // 작성
}

void readNewStudent(Student *m, int *snPtr) // 작성해서 제출
{
    // 작성
}

void calculateGrade(Student *m, int size) // 작성해서 제출
{
    // 작성
}

int main(void) // 변경하지 말라
{
    Student Students[MAX];
    int i;
    int StudentNum = 0;
    int choice;

    scanf("%d", &choice);
    while (choice != -1) {
        readNewStudent(Students, &StudentNum);
        scanf("%d", &choice);
    }
    calculateGrade(Students, StudentNum);
    for (i = 0; i < StudentNum; i++)
        printStudent(&Students[i]);
}
```

/* 문제 4.2: 구조체 2

문제4_1에서는 학생의 정보를 입력 받고 학점을 계산하였다.

문제4_1를 이용하여 가장 점수가 좋은(중간고사 + 학기말고사가 가장 높은) 학생의 정보(이름과 학점)을 출력하는 프로그램을 작성하려 한다.

bestStudent함수는 가장 점수가 좋은 학생의 인덱스를 반환한다. 함수 bestStudent를 완성하라.

실행예

```
1          <--- 새 정보 입력
ccc 30 90
1          <--- 새 정보 입력
DDD 30 50
1          <--- 새 정보 입력
fff 100 100
-1         <--- 정보 입력 끝
fff 100 100 A    <-- 출력: 가장 성적이 좋은 학생이다
```

유의사항

- main함수와 선언(include 문장, define 문장, 타입 정의 포함)을 *제외*한 사용자 정의 함수들에 대한 코드 제출

```
*/
#include <stdio.h>
#include <stdlib.h>
#define MAX 20
typedef struct {
    char name[MAX];
    int mid;
    int final;
    char grade;
} Student;
void printStudent(Student *m) //2) { /* 이전 문제에서 작성한 코드를 제출 */ }
void readNewStudent(Student *m, int *snPtr) { /* 이전 문제에서 작성한 코드를 제출 */ }
void calculateGrade(Student *m, int size) { /* 이전 문제에서 작성한 코드를 제출 */ }
int bestStudent(Student *m, int size)
{
    // 작성해서 제출
}
int main(void) // 변경하지 말라
{

    Student Students[MAX];
    int StudentNum = 0;
    int choice;

    scanf("%d", &choice);
    while (choice != -1) {
        readNewStudent(Students, &StudentNum); // 새 회원 정보를 읽어 들인다
        scanf("%d", &choice);
    }
    calculateGrade(Students, StudentNum); // 모든 학생의 학점을 계산
    printStudent(&Students[bestStudent(Students, StudentNum)]);
}
```

/*문제 4_3: 구조체 3

주어진 뼈대코드를 보면 앞의 구조체 문제 4_1에 CClass라는 구조체 타입이 추가되었다.

문제 4_1과 같은 일을 하는 프로그램을 주어진 main 함수로 이용하여 프로그램을 작성하라.

유의사항

- main 함수와 선언(include 문장, define 문장, 타입 정의 포함)을 *제외*한 사용자 정의 함수들에 대한 코드 제출.

```
*/
#include <stdio.h>
#include <stdlib.h>
#define MAX 20
typedef struct {
    char name[MAX];
    int mid;
    int final;
    char grade;
} Student;

typedef struct {
    int num;
    Student s[40];
} CClass;

// 필요한 함수들의 정의를 추가해서 이를 제출

int main(void) // 변경하지 말라
{
    CClass cClass;

    cClass.num = 0; // 초기에는 학생수 0

    readClass(&cClass);
    calculateClassGrade(&cClass); // 학점 계산
    printClass(&cClass); // 학생 정보를 출력
}
```


/*문제 5: 오목 응용

우리는 마지막 숙제였던 오목완성에서 오목 구현을 위한 winCheck함수를 작성하였다.
그 winCheck함수를 수정하여 이 문제를 풀어보라.

말놓기가 이미 진행된, 어느 정도의 판의 형태로 입력을 받는다.(main함수의 (A)부분)

말을 놓지 않은 곳은 '-'로 표시되며 플레이어 X, O가 말을 놓으면 각각 'X'와 'O'가 놓여진다.

여기에 플레이어 X와 플레이어 O가 차례로 한번씩 말을 놓으려 한다.

말을 놓았을 때 대각선, 역대각선에 그 말이 총 몇 개 있는가를 출력하려 한다.

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	X	-	-	-	-
1	-	-	-	-	X	-	-	O	O	-
2	X	O	X	-	O	O	X	O	X	O
3	-	X	X	X	-	O	X	O	O	O
4	-	-	-	-	X	-	O	-	-	-
5	X	X	-	O	-	O	X	O	X	O
6	X	X	-	O	-	O	X	O	X	O
7	X	X	X	O	-	O	X	O	X	O
8	X	X	-	O	-	O	X	O	X	O
9	X	X	-	O	X	O	O	O	X	O

즉, 위의 그림에서처럼 아래와 같은 말들이 놓여진 상태가 입력된 후
플레이어 X가 (7, 2)를 선택하면(빨간색 X)
그림에 회색으로 표시된 것 같이 대각선과 역대각선에서의 X의 개수를 세어서 이를 출력한다.
플레이어 0에 대해서도 같은 방법으로 출력한다.

실행예

```

----X----   <- 이하 10줄은 입력: 이미 놓인 말들
----X--00-
X0X-00X0X0
-XXX-0X000
----X-0---
XX-0-0X0X0
XX-0-0X0X0
XX-0-0X0X0
XX-0-0X0X0
XX-0X000X0

7 2 <- 입력: 플레이어 x가 말을 놓는다
7 <- 출력: 대각선, 역대각선상의 x의 개수

1 3 <- 입력: 플레이어 0가 말을 놓는다
6 <- 출력: 대각선, 역대각선상의 0의 개수

```

유의사항

- winCheck 함수만 제출한다.

 $\ast/$

```

#include <stdio.h>
#define BOARD_SIZE 10

int winCheck(char b[][BOARD_SIZE + 1], int r, int c) // 이 함수만 제출
{
    // 작성
}

void display(char b[][BOARD_SIZE + 1])
{
    int i, j;
    printf("      ");
    for (i = 0; i < BOARD_SIZE; i++)
        printf("%2d", i);
    printf("\n      -----Wn");
    for (i = 0; i < BOARD_SIZE; i++) {
        printf("%3d |", i);
        for (j = 0; j < BOARD_SIZE; j++)
            printf(" %c", b[i][j]);
        printf("\n");
    }
}

int main(void)
{
    char board[BOARD_SIZE][BOARD_SIZE + 1] = {' '};
    char turn = 'X';
    int r, c;
    int i;
    int lose;

    // (A)말들이 이미 놓인 판으로 입력
    for (i = 0; i < BOARD_SIZE; i++)
        scanf("%s", board[i]); // 판의 행부분을 한꺼번에 입력 받는다

    //display(board); // 보드판의 말이 놓인 것을 보려면 주석 취소
    for (i = 0; i < 2; i++) { // 이미 말들이 놓인 판에 X, O 말을 한 번씩 놓고
        // 대각선, 역대각선에 있는 자신의 말의 수를 출력한다
        //printf("Player %c(행 열):", turn);
        scanf("%d %d", &r, &c);

        //if(board[r][c] != '-') continue; // 이미 말이 놓인 자리

        board[r][c] = turn;
        //display(board); // 보드판의 말이 놓인 것을 보려면 주석 취소

        printf("%dWn", winCheck(board, r, c));
        turn = (turn == 'X' ? 'O' : 'X');
    }
}

```