

# C Programming Mentoring

---

코드 저장소 : 멘토 수업 및 공부 자료

---

<https://github.com/LeeSM0518/TIL/tree/master/C/mentoring>

## 배열

---

### 이론

---

#### 배열선언

배열을 이루는 자료형

↑  
**int arr[10];**

↓  
배열의 이름

↓  
배열의 길이

#### 1차원 배열

**int arr[3] = {10, 30, 20};**

10	30	20
----	----	----

**arr[0]**

**arr[1]**

**arr[2]**

## 2차원 배열

```
int arr[2][3]={ {10, 20, 30}, {50, 70, 60} };
```

	0열	1열	2열
0행	arr[0][0] = 10;	arr[0][1] = 20;	arr[0][2] = 30;
1행	arr[1][0] = 50;	arr[1][1] = 70;	arr[1][2] = 60;

## 실습

### 주사위 던지기

#### 중요!!

- 배열은 0부터 시작한다.
- 배열의 범위와 위치를 가리키는 [] 안에 식이 들어갈 수 있다.

```
#include <stdio.h>
#include <stdlib.h> // 랜덤함수를 써주기 위해
#include <time.h>   // 지속적으로 랜덤 값을 변경시켜 주기 위해
#define SIZE 6     // 전역 변수 선언

int main(void)
{
    int freq[SIZE] = { 0 }; // 배열을 다 0으로 초기화
    int i;

    srand((unsigned)time(NULL)); // 랜덤 함수를 지속적으로 값을 변경 시켜주기 위해

    for( i = 0 ; i < 10000 ; i ++){
        // 10000 번 주사위를 돌려준다
    }
```

```

    ++freq[rand() % 6];
    // rand() % 6 한 값은 0부터 5까지의 값이 나온다.
    // 랜덤함수에서 나온 값의 freq 배열 위치에 1을 증가시킨다.
}

printf("면\t빈도\n");

for( i = 0 ; i < SIZE ; i++){
    printf("%3d \t %3d \n", i, freq[i]);
    // %3d 는 스페이스바 2번 한 뒤에 값이 나온다.
    // \t 는 Tab키 한번을 뜻한다.
}

return 0;
}

```

## 실행결과

면	빈도
0	1671
1	1664
2	1694
3	1628
4	1675
5	1668

## 최소값, 최대값 구하기

### 중요!!

1. minimum 값과 maximum 값을 미리 prices[0] 값으로 넣지 않으면 오류가 발생한다.
2. prices[i] 값이 min 보다 작을 때 값을 바꿔주고 prices[i] 값이 max 보다 클 때 값을 바꿔주는 조건문

### • 코딩

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 10 // SIZE 라는 전역변수 선언

```

```

int main(void)
{
    int prices[SIZE] = { 0 };
    int i, minimum, maximum;

    printf("-----\n");
    printf("1  2  3  4  5  6  7  8  9  10\n");
    printf("-----\n");
    srand((unsigned)time(NULL));

    for( i = 0 ; i < SIZE ; i++){
        prices[i] = (rand() % 100) + 1;
        // prices 배열에 순서대로 0 부터 SIZE 번 까지에
        // 랜덤 함수를 이용해 1 ~ 100 까지의 값을 넣어준다.

        printf("%d ",prices[i]);
        // 값 출력

    }

    printf("\n\n");

    minimum = prices[0];
    maximum = prices[0];

    // prices 배열의 첫 번째 값을 미리 최소값과 최대값에 넣어준다.
    // ( 밑에 for문에서 값 비교를 위해서 )

    for(i = 1 ; i < SIZE ; i++)
    {
        if(prices[i] < minimum){
            minimum = prices[i];
        }
        // 저장해둔 최소값이 prices[i] 보다 크면 prices[i] 값이
        // 최소값 보다 더 작다는 것을 의미하므로 최소값을 바꿔준다.

        if(prices[i] > maximum){
            maximum = prices[i];
        }
        // 저장해둔 최대값이 prices[i] 보다 작으면 prices[i] 값이
        // 최대값 보다 더 크다는 것을 의미하므로 최대값을 바꿔준다.

    }

    printf("최소값은 %d 입니다.\n", minimum);
    printf("최대값은 %d 입니다.\n", maximum);
}

```

- 실행결과

-----  
1 2 3 4 5 6 7 8 9 10  
-----

61 3 18 3 21 24 25 94 92 98

최소값은 3 입니다.

최대값은 98 입니다.

## 영화관 좌석 예매

### 중요!!

1. 원래는 main 함수 말고 다른 함수들은 실행이 끝나는 동시에 **값이 사라지지만** 배열을 **전역변수**로 선언했기 때문에 함수를 **void 형식**으로 하고 void 함수 내에서 값을 처리하고 함수가 끝나고 **값이 사라지지 않는다**.
2. **while(1)** // 무한루프 : break 문이 없으면 { } 안의 내용이 무한적으로 반복한다.

### • 코딩

```
#include <stdio.h>
#define SIZE 10

int seats[SIZE] = { 0 }; // void 함수가 seats 배열의 값을 처리해도 값이 그대로 seats에 남는다.

void reserve1(); // 한명을 예약했을때 실행 함수

void reserve2(int count); // 두명 이상 예약했을때 실행 함수

void print_seat(); // 좌석을 출력 실행 함수

int check_full(); // 좌석이 얼마나 예약됐는지 확인 실행 함수

int main(void)
{
    char ans1; // 좌석을 예약할지 말지를 결정하기 위한 변수
    int count; // 몇 개의 좌석을 예약할지를 결정하기 위한 변수
    int i; // for 문을 위한 변수
    int check = 0; // 모든 자리가 예약됐는지 확인하기 위한 변수
    int reserved; // 몇자리가 예약됐는지 확인하기 위한 변수

    while(1)
        // 1은 반드시 참이기 때문에 while이 무한 루프가 돌기 시작한다.
    {
```

```

reserved = SIZE - check;
// SIZE 전역변수에 check를 빼면 자리가 몇 자리 남았는지 알 수 있다.

printf("좌석을 예약하시겠습니까?(y 또는 n)");
scanf(" %c", &ans1);
if(ans1 == 'y')
{
    print_seat();
    printf("몇 명을 예약하실것 입니까? ");
    scanf("%d", &count);

    // 1명 예약이고 count가 남은 자리보다 작거나 같을 때
    if(count == 1 && count <= reserved){
        reserve1();
    }

    // 2명 이상 예약이고 count가 남은 자리보다 작거나 같을 때
    else if(count <= SIZE && count <= reserved)
        reserve2(count);

    // 남은 자리가 count 보다 작을 때
    else{
        printf("자리가 부족합니다.\n");
    }
}

else if(ans1 == 'n'){
    return 0;
    // while 반복문을 나가게 된다.
}

check = check_full();
// check 변수에 몇 자리가 예약되었는지의 값을 넣어준다.

if(check == SIZE){
    // check의 값이 10 이 될때 while문을 나가게된다.
    printf("모든 자리가 예약되었습니다.\n");
    break;
}

// check 값이 아직 10이 되지 않았을 때 아무일도 일어나지 않고 계속 while문을 실행한다
else
    continue;
}

return 0;
}

void reserve1()
{
    int ans2, i;

    printf("몇 번째 좌석을 예약하시겠습니까? ");

```

```

scanf("%d", &ans2);
// ans2 에 예약할 좌석 값을 받는다

if(ans2 <= 0 || ans2 > SIZE ){
    printf("1부터 10사이의 숫자를 입력하세요\n");
}
// 좌석 위치 보다 작거나 큰 값을 입력했을 경우

if(seats[ans2-1] == 0){
    seats[ans2-1] = 1;
    printf("예약되었습니다.\n");
}
// ans2 - 1 을 하는 이유는 배열의 시작은 0 부터 이기 때문이다.

print_seat();
// 좌석들 출력.
}

void reserve2(int count)
{
    int ans2, i;
    for( i = 0 ; i < count ; i++)    // 여러명을 예약하기 위해 for문을 실행
    {
        printf("몇 번째 좌석을 예약하시겠습니까? ");
        scanf("%d", &ans2);

        if(ans2 <= 0 || ans2 > SIZE ){
            printf("1부터 10사이의 숫자를 입력하세요\n");
        }

        if(seats[ans2-1] == 0){
            seats[ans2-1] = 1;
            printf("예약되었습니다.\n");
        }

        else{
            printf("이미 예약된 자리입니다.\n");
        }
    }
    print_seat();
}

// 좌석이 얼마나 남았는지 출력해주는 함수.
// seats 를 전역변수로 선언되어있기 때문에 매개변수로 받지 않아도 된다.
void print_seat()
{
    int i;

    printf("-----\n");
}

```

```

printf("1 2 3 4 5 6 7 8 9 10\n");
printf("-----\n");

for(i = 0; i < SIZE ; i++){
    printf("%d ",seats[i]);
}
printf("\n");
}

int check_full()
{
    int i;
    int count = 0;

    for(i = 0 ; i < SIZE ; i++){
        if( seats[i] == 1 ){
            count++;
        }
    }
    // 좌석이 예약되었는지 조건문으로 확인하여 count를 1씩 증가시킨뒤 반환시킨다.
    return count;
}

```

- 실행결과

```

좌석을 예약하시겠습니까?(y 또는 n)y
-----
1 2 3 4 5 6 7 8 9 10
-----
0 0 0 0 0 0 0 0 0 0
몇 명을 예약하실것 입니까? 15
자리가 부족합니다.
좌석을 예약하시겠습니까?(y 또는 n)y
-----
1 2 3 4 5 6 7 8 9 10
-----
0 0 0 0 0 0 0 0 0 0
몇 명을 예약하실것 입니까? 10
몇 번째 좌석을 예약하시겠습니까? 1
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 2
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 3
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 4
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 5
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 6
예약되었습니다.
몇 번째 좌석을 예약하시겠습니까? 7

```



예약되었습니다.  
몇 번째 좌석을 예약하시겠습니까? 8  
예약되었습니다.  
몇 번째 좌석을 예약하시겠습니까? 9  
예약되었습니다.  
몇 번째 좌석을 예약하시겠습니까? 10  
예약되었습니다.

-----  
1 2 3 4 5 6 7 8 9 10  
-----

1 1 1 1 1 1 1 1 1 1

모든 자리가 예약되었습니다.

## 2차원 배열의 합계, 최대값, 최소값

### 중요!!

1. 최대값은 가장 작은 값으로 초기화 시키고 최소값은 가장 큰 값으로 초기화한다.
2. 가로줄을 행, 세로줄을 열 이라 한다.
3. 바깥에 있는 for문과 안에 있는 for문 구분을 잘해야한다.
  - 밖의 for문이 시작 >> 안에 있는 for문이 다 실행을 마치고 >> 밖에 있는 for문이 실행

### • 코딩

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

#define SIZE 100

int main(void)
{
    int i, j;
    int box[3][5] = { 0 }; // 2차원 배열 선언

    int max = -1, min = SIZE + 1;
    // box 배열에 들어갈 값이 1 ~ 100 이기 때문에
    // 최소를 SIZE + 1, 최대를 -1 로 초기화해야만
    // 처음에 값을 비교할때 더 크거나 작은 값을 받아들일수있다.

    int sum_row[3] = {0}; // 각 행의 총합을 계산하기 위함
```

```

int sum_col[5] = {0};    // 각 열의 총합을 계산하기 위함

srand((unsigned)time(NULL));

printf("\t숫자\t숫자\t숫자\t숫자\t숫자\t|\t\t합계\t최대값\t최소값\n");

for( i = 0 ; i < 3 ; i++){    // 행이 3줄이므로 3번 반복

    printf("숫자");

    for( j = 0 ; j < 5 ; j++){    // 열이 5줄이므로 5번 반복

        box[i][j] = rand() % SIZE + 1;    // 1 ~ 100 까지의 값을 랜덤으로 넣는다.
        printf("\t%d", box[i][j]);    // 랜덤 값 출력

        if(box[i][j] > max)        // 만약 i열 j행의 값이 max 보다 크면 max의 값을 바꿈.
            max = box[i][j];

        if(box[i][j] < min)        // 만약 i열 j행의 값이 min 보다 작으면 min의 값을 바꿈.
            min = box[i][j];

        sum_row[i] += box[i][j];    // i열의 배열을 모두 더한다.

    }

    printf("\t|");

    printf("\t%d", sum_row[i]);    // i열 총합 출력
    printf("\t%d", max);        // 최대값 출력
    printf("\t%d", min);        // 최소값 출력

    max = -1;                    // 최대값에 i열의 최대값이 저장되어 있으므로 다시 -1로 초기화
    min = SIZE + 1;              // 최소값에 i열의 최소값이 저장되어 있으므로 다시 SIZE + 1로 초기화

    printf("\n");

}    // for문 종료

printf("-----\n");

printf("합계\t");
for( i = 0 ; i < 5 ; i++ ){    // 열이 5줄 이므로 5번 반복

    for( j = 0 ; j < 3 ; j ++){    // 행이 3줄 이므로 3번 반복

        sum_col[i] += box[j][i];
        // 각 i열의 총합을 구한다.

    }

    printf("%d\t", sum_col[i]);

```

```

}    // for문 종료

printf("\n");

max = -1;
min = SIZE + 1;

printf("최대값\t");
for( i = 0 ; i < 5 ; i++ ){
    for( j = 0 ; j < 3 ; j ++){
        if( box[j][i] > max )
            max = box[j][i];
        if( box[j][i] < min )
            min = box[j][i];
    }
    printf("%d\t",max);
    max = - 1;
    min = SIZE + 1;
}
printf("\n");

printf("최소값\t");
for( i = 0 ; i < 5 ; i++ ){
    for( j = 0 ; j < 3 ; j ++){
        if( box[j][i] > max )
            max = box[j][i];
        if( box[j][i] < min )
            min = box[j][i];
    }
    printf("%d\t",min);
    max = - 1;
    min = SIZE + 1;
}

printf("\n");
}

```

- 실행결과

	숫자	숫자	숫자	숫자	숫자		합계	최대값	최소값
숫자	64	74	60	3	45		246	74	3
숫자	41	59	78	10	39		227	78	10
숫자	67	86	44	60	90		347	90	44
<hr/>									
합계	172	219	182	73	174				
최대값	67	86	78	60	90				
최소값	41	59	44	3	39				

