



**ROBIT**  
ROBOT SPORT GAME TEAM

# C Language 기초 - 3일차

1. 동적할당
2. 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

동적할당

# 동적할당



**ROBIT**  
ROBOT SPORT GAME TEAM

- 정적인 방식 :  
프로그램 실행 전 **미리 정해진 크기의 메모리**를 먼저 할당 후 프로그램 종료 시 해제  
ex) `int A[100];`
- 동적인 방식 :  
프로그램 실행 도중에 **필요한 만큼의 메모리** 할당 후 원할 때 해제  
ex) `A = (int *)malloc(100*sizeof(int));`

# 동적할당



**ROBIT**  
ROBOT SPORT GAME TEAM

함수	기능
<code>void * malloc ( size_t size );</code>	size 바이트의 메모리를 힙에서 할당하여 반환한다.
<code>void * calloc ( size_t num, size_t size );</code>	(num * size) 바이트의 메모리를 힙에서 할당하고 포인터값을 반환한다.
<code>void * realloc ( void * ptr, size_t size );</code>	ptr이 가리키는 메모리를 size 바이트만큼 힙에서 재할당하여 반환한다.
<code>void free ( void * ptr );</code>	ptr이 가리키는 메모리를 해제한다. 해제 전까지 계속 존재하므로 필요없으면 이 함수에 의해 해제해야 한다.

- 위와 같은 함수를 사용하기 위해 std 라이브러리를 추가해준다.
- `#include <stdlib.h>` 추가

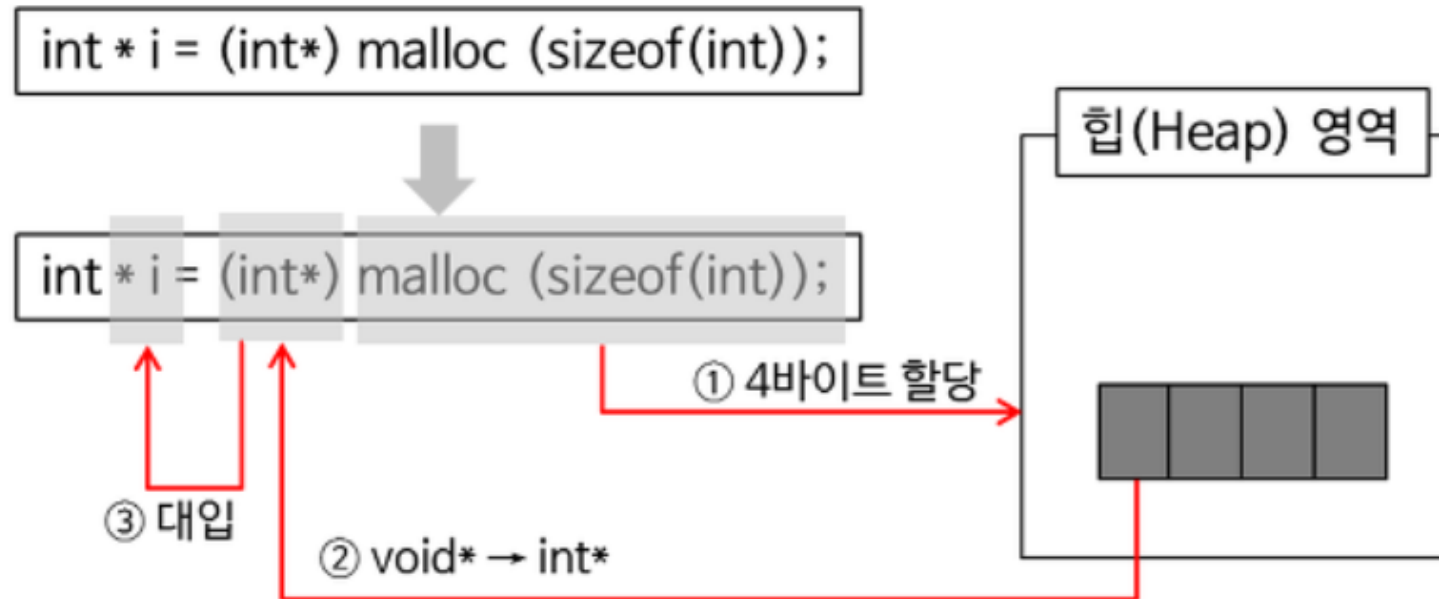


# 동적할당 \_malloc함수

- malloc 함수는 동적 메모리를 할당하는 함수이다.

Ex) `int *i = (int*) malloc (sizeof(int));`

할당을 받고자 하는 바이트 수





# 동적할당 \_free함수

- free함수는 동적 메모리를 반납하는 함수이다. (할당된 메모리 해제)
- 동적할당 후 해당 변수를 더 사용할 필요가 없다면 반드시 **free 함수로 메모리를 해제**

`int *arr = (int*) malloc(sizeof(int)*input);` —————> 동적 메모리 할당

`free(arr_2);` —————> 동적 메모리 해제

# 동적할당 \_예제



**ROBIT**  
ROBOT SPORT GAME TEAM

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

int main(void)
{
    int arr[4] = {4,3,2,1};
    int* pArr;

    pArr = (int*)malloc(sizeof(int)*4);

    if(pArr == NULL)
    {
        printf("malloc error");
    }
}
```

```
for(int i=0; i<4; ++i)
{
    pArr[i] = arr[i];
}

for(int i=0; i<4; ++i)
{
    printf("%d\\n", pArr[i]);
}

free(pArr);

return 0;
}
```

```
4
3
2
1
계속하려면 아무 키나 누르십시오 . . .
```



# 동적할당 \_calloc함수

- calloc 함수는 "0으로 초기화된" 동적 메모리를 할당한다.

`int *p = (int *)calloc(5, sizeof(int));`

→ 항목의 개수

→ 항목의 크기





# 동적할당 \_realloc함수

- realloc 함수는 할당하였던 메모리의 블록의 크기를 변경한다.

```
int *p = (int *)malloc( 5*sizeof(int) );
```

```
p=realloc( p ,7*sizeof(int) );
```

↓  
기존 동적메모리

↘ 새롭게 할당하고자 하는 메모리 크기

# 동적할당 \_예제



**ROBIT**  
ROBOT SPORT GAME TEAM

```
int main()
{
    printf("정수 2개를 저장할 공간이 필요 \n");
    int* list = (int*)malloc(sizeof(int) * 2);
    if (list == NULL)
    {
        printf("동적 메모리 할당 오류\n");
        exit(1);
    }

    list[0] = 10;
    list[1] = 20;
```

```
    printf("정수 3개를 저장할 공간으로 확장 \n");
    int* list_new = (int*)realloc(list, sizeof(int) * 3);
    if (list_new == NULL)
    {
        printf("동적 메모리 할당 오류\n");
        exit(1);
    }

    list_new[2] = 30;

    for (int i = 0; i < 3; i++)
        printf("%d ", list_new[i]);
    printf("\n");

    free(list_new);

    return 0;
}
```



**ROBIT**  
ROBOT SPORT GAME TEAM

# 다차원 배열

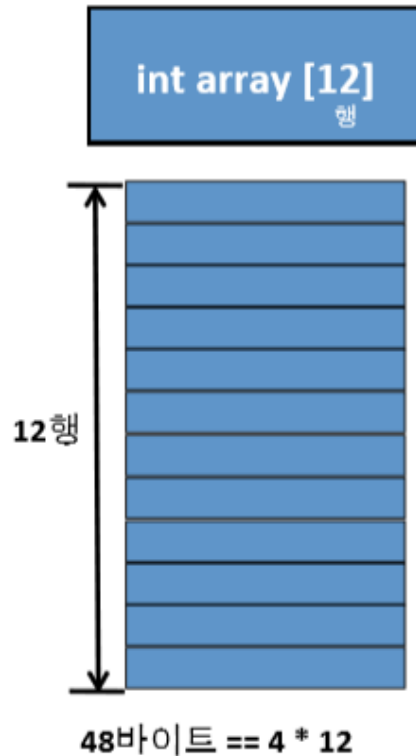
# 다차원 배열



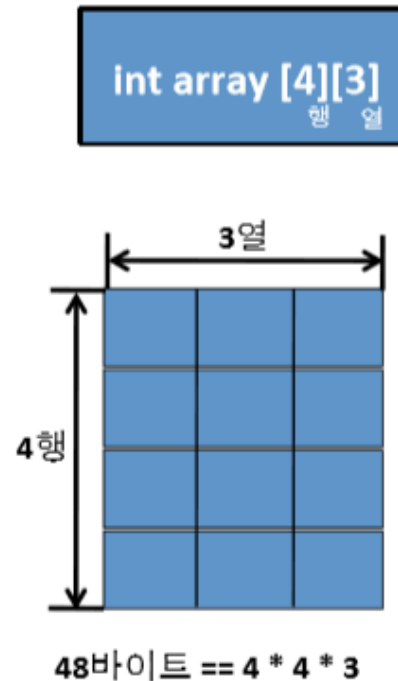
**ROBIT**  
ROBOT SPORT GAME TEAM

- 2차원 이상의 배열을 의미

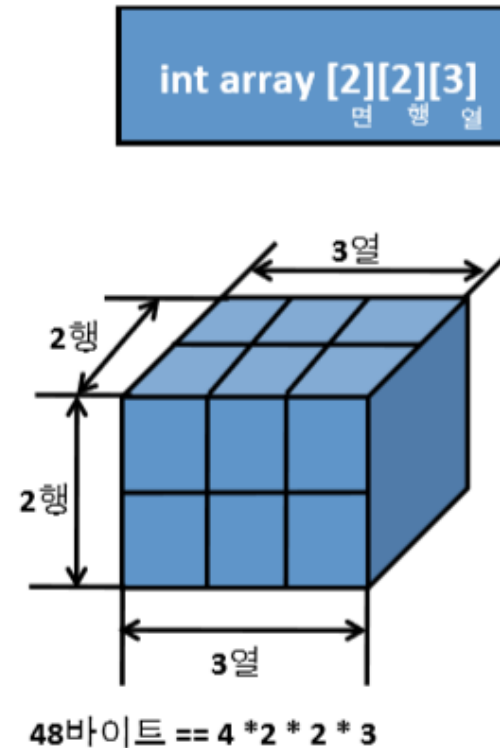
1차원 배열



2차원 배열



3차원 배열



# 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

- 자료형 : 배열의 자료형을 지정
- 배열 이름 : 변수 이름과 마찬가지로 배열을 구분하는 배열의 이름
- 배열 길이 : 배열 요소의 길이를 행(가로)과 열(세로)로 지정

자료형      배열 이름      [배열 길이]

↓            ↓            ↓

int   array[4][3];

                          ↓        ↓

                          행      열

# 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include<stdio.h>
int main(void)
{
    //2차원 배열의 선언
    int array[4][3];

    //4행 3열의 배열 길이 선언
    array[0][0]=1; array[0][1]=2; array[0][2]=3;
    array[1][0]=4; array[1][1]=5; array[1][2]=6;
    array[2][0]=7; array[2][1]=8; array[2][2]=9;
    array[3][0]=10; array[3][1]=11; array[3][2]=12;

    printf("%d %d %d\\n", array[0][0], array[0][1], array[0][2]); //0행 출력
    printf("%d %d %d\\n", array[1][0], array[1][1], array[1][2]); //1행 출력
    printf("%d %d %d\\n", array[2][0], array[2][1], array[2][2]); //2행 출력
    printf("%d %d %d\\n", array[3][0], array[3][1], array[3][2]); //3행 출력

    return 0;
}
```



# 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

```
#include<stdio.h>
int main(void)
{
    //2차원 배열의 선언
    int array1[4][3] = {1,2,3,4,5,6,7,8,9,10,11,12};
    int array2[4][3] = {1,2,3,4,5};

    //2차원 배열 array1의 데이터 출력
    printf("%d %d %d\n", array1[0][0], array1[0][1], array1[0][2]); //0행 출력
    printf("%d %d %d\n", array1[1][0], array1[1][1], array1[1][2]); //1행 출력
    printf("%d %d %d\n", array1[2][0], array1[2][1], array1[2][2]); //2행 출력
    printf("%d %d %d\n", array1[3][0], array1[3][1], array1[3][2]); //3행 출력

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

    //2차원 배열 array2의 데이터 출력
    printf("%d %d %d\n", array2[0][0], array2[0][1], array2[0][2]); //0행 출력
    printf("%d %d %d\n", array2[1][0], array2[1][1], array2[1][2]); //1행 출력
    printf("%d %d %d\n", array2[2][0], array2[2][1], array2[2][2]); //2행 출력
    printf("%d %d %d\n", array2[3][0], array2[3][1], array2[3][2]); //3행 출력

    return 0;
}
```

int array1[4][3]

1	2	3
4	5	6
7	8	9
10	11	12

int array2[4][3]

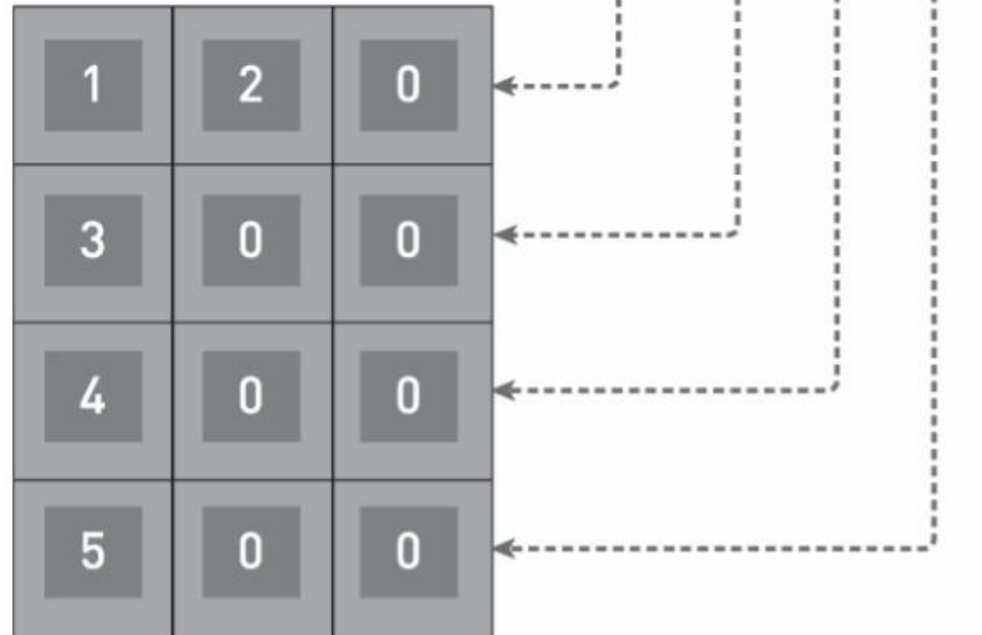
1	2	3
4	5	0
0	0	0
0	0	0

# 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

```
int array1[4][3]={ {1, 2}, {3}, {4}, {5}}
```





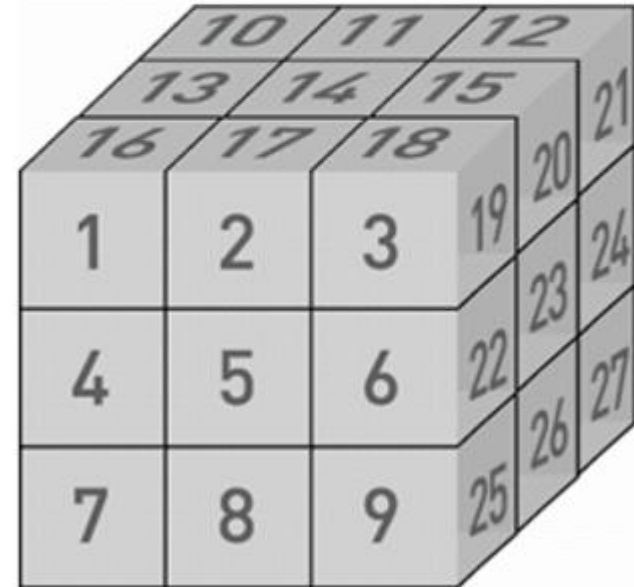
# 다차원 배열



**ROBIT**  
ROBOT SPORT GAME TEAM

```
//2차원 배열에 데이터 출력
for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        printf("%3d",array[i][j]);
    }
    printf("\n");
}
```

면    행    열  
↓    ↓    ↓  
int array[3][3][3]





**ROBIT**  
ROBOT SPORT GAME TEAM

# 과제

# 과제 관련 주의사항



**ROBIT**  
ROBOT SPORT GAME TEAM

1. 제출 기한 : 다음 날 수업 전까지
2. 제출 형식 : 프로젝트 파일, 보고서(PDF) 를 (로빗\_18기\_수습단원\_이름) 으로 압축 후  
[kwrobit2023@gmail.com](mailto:kwrobit2023@gmail.com) 으로 제출  
(보고서는 코드 설명과 실행 화면 첨부, 메일 제목은 C언어\_n일차\_이름)
3. 채점 기준 :
  - 1) 프로그램의 실행가능 여부
  - 2) 교육하지 않은 C언어 개념 사용 시 감점
  - 3) 예외처리
  - 4) 효율적인 코드 작성
  - 5) 제출 형식

# 과제 1



**ROBIT**  
ROBOT SPORT GAME TEAM

행렬 5X5를 이용하여 1씩 증가시켜 아래 사진과 같이 모래시계를 만드시오.  
(빈칸은 0으로 채우면 됩니다)

1	2	3	4	5
	6	7	8	
		9		
	10	11	12	
13	14	15	16	17

# 과제 2



**ROBIT**  
ROBOT SPORT GAME TEAM

N1XN2

아래 그림과 같이 2차원 배열에 숫자를 채운 후 (배열A) 행과 열을 바꿔 출력 (배열B) 하시오.

1	2	3
4	5	6
7	8	9
10	11	12

배열A



1	4	7	10
2	5	8	11
3	6	9	12

배열B

# 과제 3



**ROBIT**  
ROBOT SPORT GAME TEAM

N1XN2행렬을 입력 받아 아래 사진과 같이 왼쪽 위부터 1로 시작하여 아래와 같은 규칙을 가지도록 프로그램을 하시오. (동적할당 사용하시길 바랍니다)

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

# 과제 4



**ROBIT**  
ROBOT SPORT GAME TEAM

동적할당을 이용하여 배열의 값들의 최대, 최소, 전체 합, 평균을 구하시오.

```
몇 개의 원소를 할당하겠습니까? : 7
정수형 데이터 입력:1
정수형 데이터 입력:2
정수형 데이터 입력:3
정수형 데이터 입력:4
정수형 데이터 입력:5
정수형 데이터 입력:6
정수형 데이터 입력:7
최대값: 7
최소값: 1
전체합: 28
평균: 4.000000
계속하려면 아무 키나 누르십시오 . . .
```

# 과제 5



**ROBIT**  
ROBOT SPORT GAME TEAM

동적할당으로  $N \times N$ 을 할당 받아 오른쪽 그림과 같이 대각선으로 숫자를 채우는 프로그램을 작성하시오.

1	2	4	7	11
3	5	8	12	16
6	9	13	17	20
10	14	18	21	23
15	19	22	24	25