



ROBIT
ROBOT SPORT GAME TEAM

C Language 기초 - 2일차

1. 함수
2. 형 변환
3. 배열



ROBIT
ROBOT SPORT GAME TEAM

기초 이론

scanf 사용



ROBIT
ROBOT SPORT GAME TEAM

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

int main(void)
{
    return 0;
}
```

#include <stdio.h> 앞에 #define _CRT_SECURE_NO_WARNINGS 를
적어주어 scanf_s가 아닌 scanf 를 사용

아스키코드

• 아스키코드

- 아스키코드는 숫자로 문자를 표현하기 위한 약속
- 특수문자, 숫자, 문자에 번호를 부여하여 컴퓨터에서 처리하기 쉽게 만드는 것이다.
- 아스키코드는 미국표준협회(ANSI)에서 지정한 표준부호이다.

+) Dec : 정수 표현

Hex : 16진수 표현



ROBIT
ROBOT SPORT GAME TEAM

Dec	Hex	문자	Dec	Hex	문자	Dec	Hex	문자	Dec	Hex	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	`
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i
10	0x0a	LF	42	0x2a	*	74	0x4a	J	106	0x6a	j
11	0x0b	VT	43	0x2b	+	75	0x4b	K	107	0x6b	k
12	0x0c	FF	44	0x2c	,	76	0x4c	L	108	0x6c	l
13	0x0d	CR	45	0x2d	-	77	0x4d	M	109	0x6d	m
14	0x0e	SO	46	0x2e	.	78	0x4e	N	110	0x6e	n
15	0x0f	SI	47	0x2f	/	79	0x4f	O	111	0x6f	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1a	SUB	58	0x3a	:	90	0x5a	Z	122	0x7a	z
27	0x1b	ESC	59	0x3b	;	91	0x5b	[123	0x7b	{
28	0x1c	FS	60	0x3c	<	92	0x5c	\	124	0x7c	
29	0x1d	GS	61	0x3d	=	93	0x5d]	125	0x7d	}
30	0x1e	RS	62	0x3e	>	94	0x5e	^	126	0x7e	~
31	0x1f	US	63	0x3f	?	95	0x5f	_	127	0x7f	DEL



ROBIT
ROBOT SPORT GAME TEAM

함수

함수



ROBIT
ROBOT SPORT GAME TEAM

- 함수

특정 작업을 수행하는 코드의 모임에 이름을 붙인 것이다.

함수는 필요한 데이터를 전달받을 수 있고, 작업의 결과를 호출자에게 반환할 수 있다.

함수는 반드시 **호출에 의해서** 함수 안의 작업 내용이 실행된다.

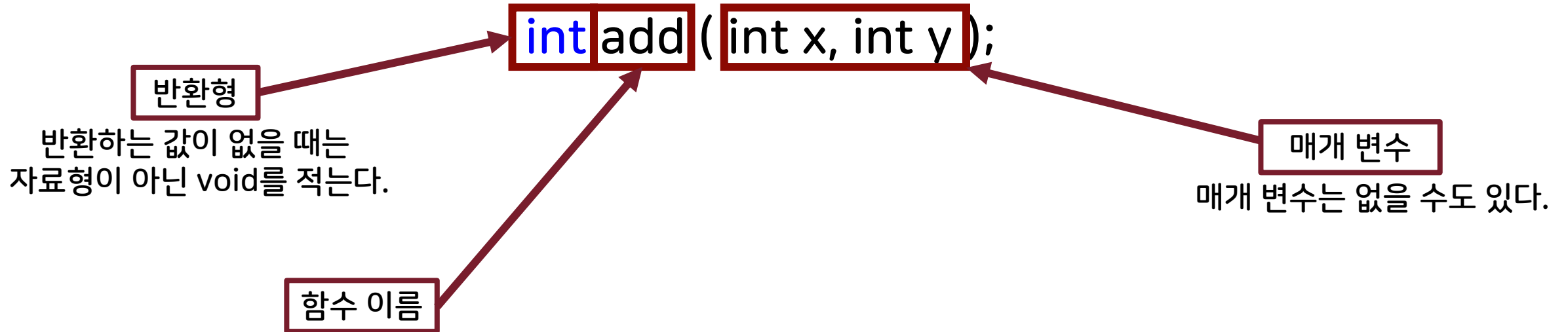
- 함수의 장점

1. 코드의 재사용 가능 (효율적)
2. 가독성 증가
3. 유지관리

함수 선언



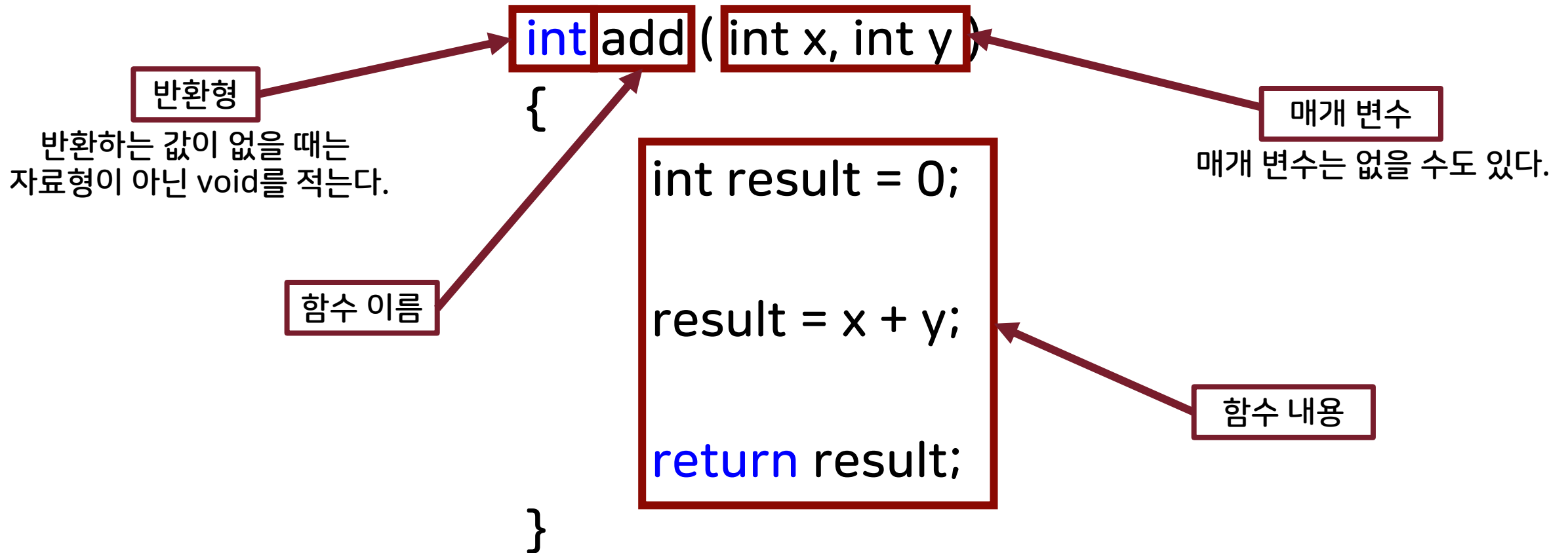
ROBIT
ROBOT SPORT GAME TEAM



함수 정의



ROBIT
ROBOT SPORT GAME TEAM



함수



RO:BIT
ROBOT SPORT GAME TEAM

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

```
int add(int x, int y);
```

함수 선언

```
void main()
```

```
{
```

```
    int a = 5;
```

```
    int b = 3;
```

```
    int answer = 0;
```

```
    answer = add(a,b);
```

함수 호출

```
}
```

```
int add(int x, int y)
```

```
{
```

```
    int result = 0;
```

```
    result = x + y;
```

```
    return result;
```

```
}
```

함수 정의



함수 예제1

```
03 int SumTwo(int a, int b);           // 함수 선언
04
05 int main(void)
06 {
07     int x = 10, y = 5;
08     int value;
09
10     value = SumTwo(x, y);           // 함수 호출
11
12     printf("두 수의 합 : %d\\n", value);
13
14     return 0;
15 }
16
17 int SumTwo(int a, int b)           // 함수 정의
18 {
19     int result;
20     result = a + b;
21     return result;
22 }
```

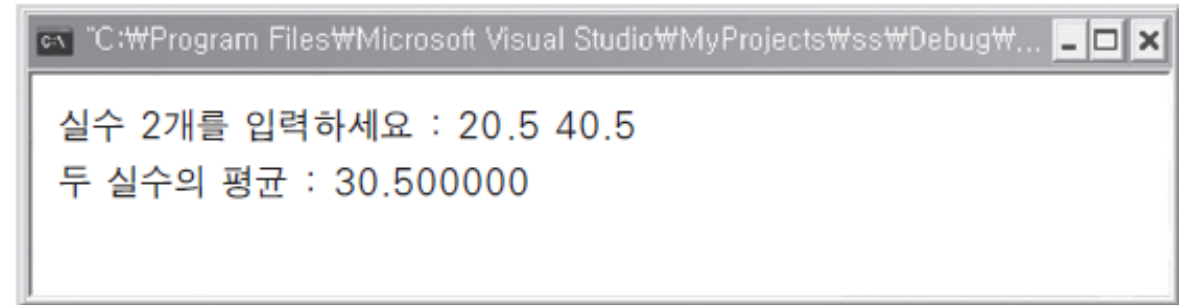


함수 예제2



ROBIT
ROBOT SPORT GAME TEAM

```
03 float Average(float x, float y);    // 함수 선언
04
05 int main(void)
06 {
07     float num1, num2, result;
08     printf("실수 2개를 입력하세요 : ");
09     scanf_s("%f %f", &num1, &num2);
10
11     result = Average(num1, num2);    // 함수 호출
12     printf("두 실수의 평균 : %f\n", result);
13     return 0;
14 }
15
16 float Average(float x, float y)    // 함수 정의
17 {
18     float z;
19     z = (x + y) / 2;
20     return z;
21 }
```





재귀함수

- 재귀호출(recursion)
 - 함수가 자기 자신을 다시 호출하는 것
 - 데이터 N개로 이뤄진 문제가 N-1개 문제를 해결하면 간단하게 해결될 때 사용
- N! 수식 표현

$$N! = N * (N-1) * (N-2) * (N-3) * \dots * 1$$

$$N! = N * (N-1)!$$

$$(N-1)! = (N-1) * (N-2)!$$

⋮

재귀함수 예제



ROBIT
ROBOT SPORT GAME TEAM

```
03 int factorial(int n);
04 int main(void)
05 {
06     int fact_num;
07     fact_num = factorial(10);
08
09     printf("10 팩토리얼 : %d\n", fact_num);
10
11     return 0;
12 }
13
14 int factorial(int n)
15 {
16     if(n <= 1)
17         return(1);
18     else
19         return(n * factorial(n-1));
20 }
```





지역 변수와 전역변수

- 지역 변수 (local variable)
 - 함수 또는 블록 안에서 정의되는 변수
 - 지역 변수는 해당블록이나 함수 안에서만 사용이 가능하다.
- 전역 변수 (global variable)
 - 함수의 외부에서 선언되는 변수
 - 전역 변수는 소스파일의 어느 곳에서도 사용이 가능하다.

전역변수 사용법



ROBIT
ROBOT SPORT GAME TEAM

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

```
int A;
```

전역변수는 자동으로
0으로 초기화 된다

```
void main() {
```

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

```
    A = 10;
```

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

```
    return 0;
```

```
}
```

실행결과

```
0
10
```

전역변수 사용법



ROBIT
ROBOT SPORT GAME TEAM

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

```
int A;
```

전역변수는 자동으로
0으로 초기화 된다

```
void main() {
```

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

```
    A = 10;
```

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

```
    return 0;
```

```
}
```

실행결과

```
0
10
```


전역변수 사용법



ROBIT
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
int A;

void main() {
    int A = 10;
    printf("%d \n",A);
    return 0;
}
```

실행결과

10

- 전역 변수와 지역 변수가 같은 이름을 사용하고 있는 경우, 지역 변수가 전역 변수에 비해 우선권을 가진다.



ROBIT
ROBOT SPORT GAME TEAM

형 변환

형 변환



ROBIT
ROBOT SPORT GAME TEAM

자료형			설명	바이트 수	범위
정수형	부호 있음	short	short형 정수	2	$-2^{15} \sim (2^{15})-1$
		int	정수	4	$-2^{31} \sim (2^{31})-1$
		long	long형 정수	4	$-2^{31} \sim (2^{31})-1$
	부호 없음	unsigned short	부호 없는 short형 정수	2	$0 \sim (2^{16})-1$
		unsigned int	부호 없는 정수	4	$0 \sim (2^{32})-1$
		unsigned long	부호 없는 long형 정수	4	$0 \sim (2^{32})-1$
문자형	부호 있음	char	문자 및 정수	1	$-2^7 \sim (2^7)-1$
	부호 없음	unsigned char	문자 및 부호 없는 정수	1	$0 \sim (2^8)-1$
부동 소수점형		float	단일 정밀도 부동 소수점	4	$1.2\text{E}-38 \sim 3.4\text{E}38$
		double	두배 정밀도 부동 소수점	8	$2.2\text{E}-308 \sim 1.8\text{E}308$

형 변환



ROBIT
ROBOT SPORT GAME TEAM

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

```
void main() {
```

```
    int a;
```

```
    double b;
```

```
    b = 2.4;
```

```
    a = b;
```

```
    printf("%d", a);
```

```
}
```

: warning C4244: '=' : conversion from 'double' to 'int', possible loss of data

형 변환

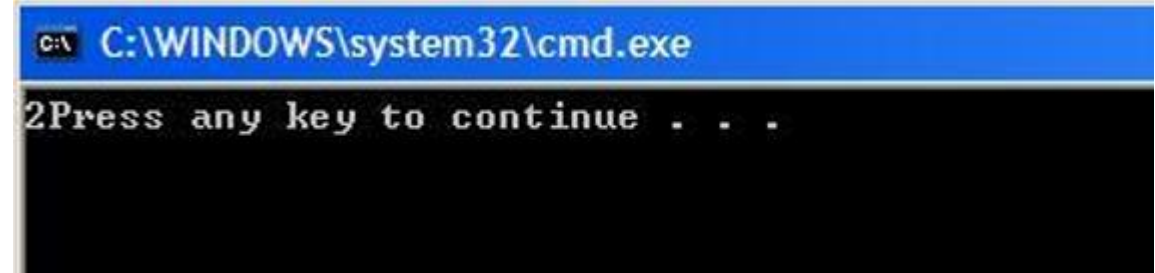


ROBIT
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
void main() {
    int a;
    double b;

    b = 2.4;
    a = (int)b;

    printf("%d", a);
}
```



(바꾸려는 자료형) 변수 이름

형 변환_예제



ROBIT
ROBOT SPORT GAME TEAM

```
#include <stdio.h>

int main() {
    int a,b;
    float c, d;

    printf("두 숫자 입력 : ");
    scanf("%d %d", &a, &b);

    c = a/b;
    d = (float)a/b;

    printf("두 수의 비율 : %f %f", c, d);
}
```

```
두 숫자 입력 : 5 2
두 수의 비율 : 2.000000 2.500000
```



ROBIT
ROBOT SPORT GAME TEAM

배열(Array)



배열(Array)

- 배열이란 무엇인가?

동일한 타입의 데이터가 여러 개 저장 되어 있는 데이터 저장 장소

- 배열은 왜 필요한가

1. 프로그램의 가독성
2. 효율적인 메모리 관리
3. 유지보수가 쉽다.



배열(Array)

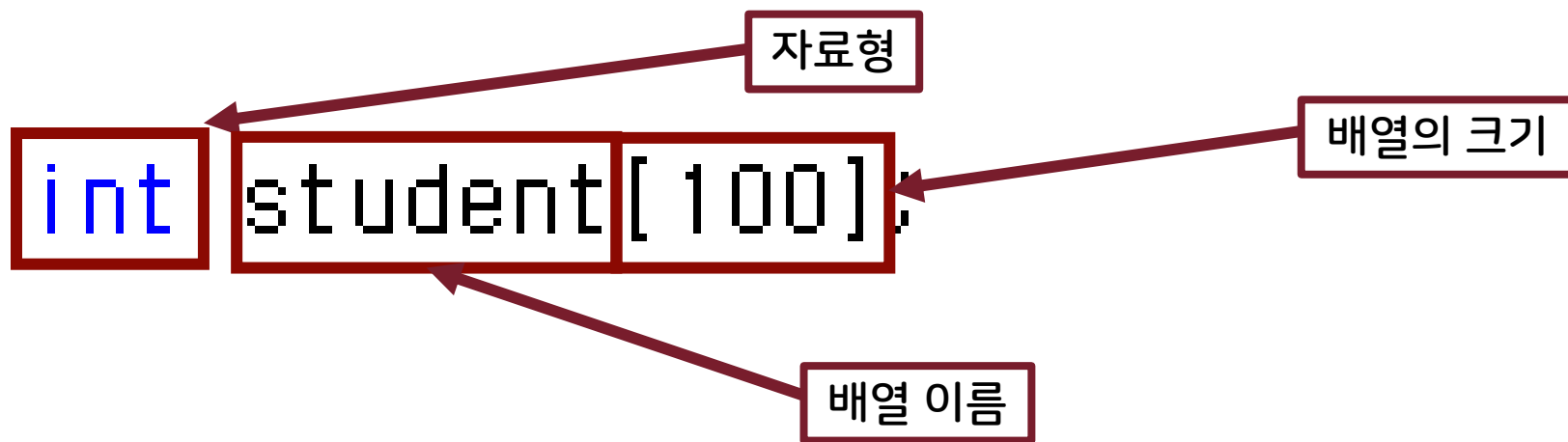
```
int student1 = 60;  
int student2 = 54;  
int student3 = 99;  
int student4 = 100;  
⋮  
int student99 = 4;
```



```
int student[100];
```



배열 선언



- 자료형 : 배열 원소들이 int형이라는 것을 의미
- 배열의 이름 : 배열을 사용할 때 사용하는 이름
- 배열의 크기 : 배열 원소의 개수를 지정해준다.
- index는 항상 0부터 시작한다.



배열 사용시 주의사항

Index가 배열의 크기를 벗어나지 않는다.

```
int i=0;  
int student[3] = {0,};
```

```
student[0] = 30;  
student[1] = 80;  
student[2] = 50;
```

```
student[3] = 10;
```

Index 값은 0과 양의 정수로 이루어진다.

```
int i=0;  
int student[3] = {0,};
```

```
student[0] = 30;  
student[1] = 80;  
student[2] = 50;
```

```
student[-1] = 10;
```

오류 발생

배열 사용



ROBIT
ROBOT SPORT GAME TEAM

```
int i=0;  
int student[5] = {0, };
```

```
student[0] = 20;  
student[1] = 40;  
student[2] = 70;  
student[3] = 40;  
student[4] = 15;
```

```
for( i=0; i<5; i++)  
{  
    printf("grade[%d] : %d\n",i,student[i]);  
}
```

```
grade[0] : 20  
grade[1] : 40  
grade[2] : 70  
grade[3] : 40  
grade[4] : 15  
계속하려면 아무 키나 누르십시오 . . .
```



문자열

- 문자열은 배열의 한 종류

```
char str[6] = "Hello";
```

```
str[0] = 'H';
```

```
str[1] = 'e';
```

```
str[2] = 'l';
```

```
str[3] = 'l';
```

```
str[4] = 'l';
```

```
str[5] = NULL;
```

배열 실습_예제



ROBIT
ROBOT SPORT GAME TEAM

데이터 탐색

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

```
void main()  
{  
    int i=0;  
    int key=0;  
    int list[8] = {1,2,3,4,5,6,7,8};  
  
    printf("탐색할 값을 적으시오 : ");  
    scanf("%d",&key);  
  
    for(i=0; i<8; i++)  
    {  
        if(list[i] == key)  
        {  
            printf("Find!! list[%d]\n",i,list[i]);  
            break;  
        }  
    }  
    printf("end\n");  
}
```

```
탐색할 값을 적으시오 : 5  
Find!! list[4]  
end  
계속하려면 아무 키나 누르십시오 . . .
```



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 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

과제1



ROBIT
ROBOT SPORT GAME TEAM

<pre> * * * * * * * * *</pre>	<pre> * * * * * * * * *</pre>	<pre> * * * * * * * * * * * * * * * * *</pre>	<pre> * * * * * * * * * * *</pre>
종류1	종류2	종류3	종류4

입력 받은 종류와 숫자를 이용하여 별을 출력하는 프로그램
(조건 : 함수를 이용하여 작성하시오)

입력1 : 길이
입력2 : 종류

```
사이즈와 종류를 입력하시오.5 1
*
**
***
**
*
```

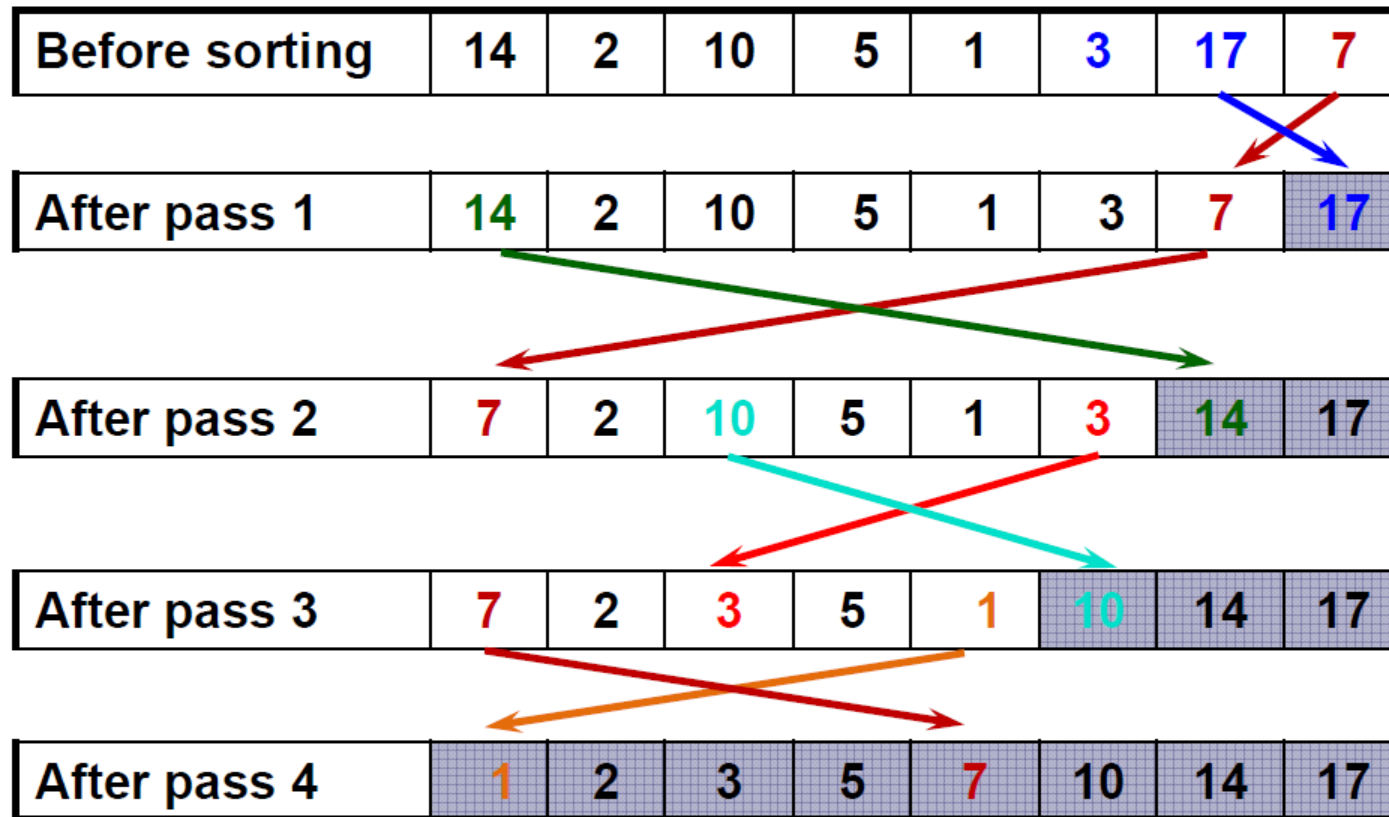
```
사이즈와 종류를 입력하시오.7 4
*****
***
**
*
**
***
*****
```



과제 2 _선택정렬

선택 정렬

- 정렬이 안된 숫자들 중에서 최대값을 선택하여 배열의 마지막 값과 교환하는 방식



과제 2 _선택정렬



ROBIT
ROBOT SPORT GAME TEAM

임의의 수를 입력받아 배열에 저장하고, 저장된 배열 안의 수를 정렬하는 프로그램을 작성하시오.
(입력은 정수 8개로 제한, 정렬이 되는 중간 과정 출력)

입력

0 : 14

1 : 2

2 : 10

3 : 5

4 : 1

5 : 3

6 : 17

7 : 7

14 2 10 5 1 3 17 7

14 2 10 5 1 3 7 17

7 2 10 5 1 3 14 17

7 2 3 5 1 10 14 17

1 2 3 5 7 10 14 17

(sorting 이 되는 중간 과정을 출력)

출력

0 : 1

1 : 2

2 : 3

3 : 5

4 : 7

5 : 10

6 : 14

7 : 17

과제 3 _음계



ROBIT
ROBOT SPORT GAME TEAM

다장조는 c d e f g a b C, 총 8개 음으로 이루어져 있다.

c부터 C까지 차례대로 연주한다면 ascending, C부터 c까지 차례대로 연주한다면 descending, 둘 다 아니라면 mixed 이다.

연주한 순서가 주어졌을 때, ascending, descending, mixed를 판별하는 프로그램을 작성하시오.
(조건 : 반환 함수를 사용 하여 반환 값에 따른 switch문을 이용)

```
음 입력  
cdefgabC
```

```
결과 : ascending
```



ROBIT
ROBOT SPORT GAME TEAM

과제 4 _문자열에서 문자열 찾기

```
최대 99개 문자 입력 <inputStr> : gewsgwghwb  
찾는 문자열 <subStr> : gh  
gh의 위치 : 7계속하려면 아무 키나 누르십시오 . . .
```

임의의 문자를 입력하고, 찾는 문자열을 입력하여, 입력 문자에서의 위치를 출력하는 프로그램을 작성하시오.

(입력 문자 안에 찾는 문자가 여러 개 있는 경우는 위치를 모두 찾아 출력)

과제 5



ROBIT
ROBOT SPORT GAME TEAM

1부터 N까지 자연수 N개로 이루어진 수열 A가 있다.
각 자연수는 수열에서 한 번씩만 등장한다.
A의 부분 수열은 A의 앞과 뒤에서 숫자를 제거해서 만들 수 있다.
길이가 홀수이면서 중앙값이 B인 A의 부분 수열의 개수를 구하는
프로그램을 작성하시오.
중앙값이란 크기 순으로 수열을 정렬했을 때, 가운데에 있는 숫자이다.
예를 들어, {5, 1, 3}의 중앙값은 3이다.

```
N : 7
B : 4
5 7 2 4 3 1 6
중앙값이 4인 부분수열의 개수 : 4
계속하려면 아무 키나 누르십시오 . . .
```

예제 입출력