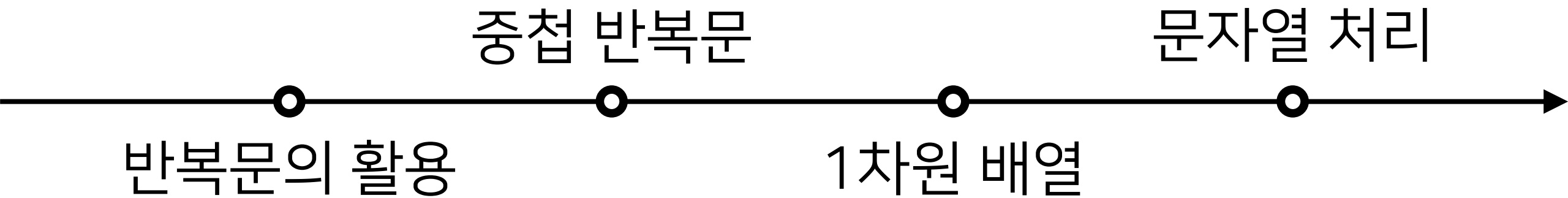


EC C언어 스터디

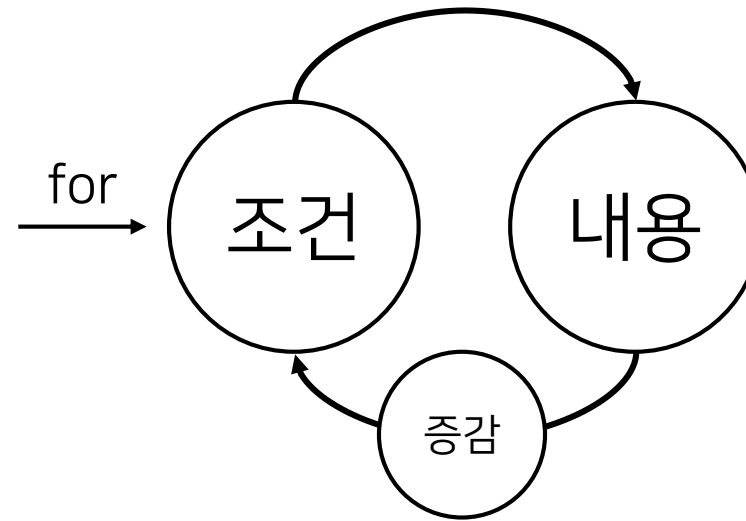
-3강-



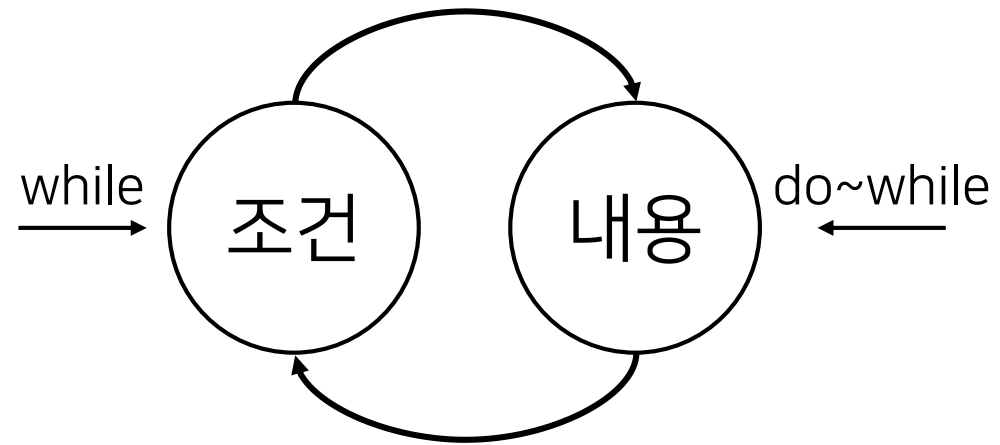


반복문의 활용

```
for(선언;조건;증감)  
{  
    내용  
}
```



```
while(조건)
{
    내용
}
```



```
do
{
    내용
} while(조건);
```

- while문 : 단순 반복
- for문 : 반복 + 인덱싱

do~while문의 활용 예제

- 반복문의 활용

```
#include <stdio.h>

int main()
{
    int n;
    do
    {
        printf("1 과 10 사이의 정수를 입력해 주세요 : ");
        scanf("%d", &n);
    } while (n <= 0 || n > 10);

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

```
#include <stdio.h>

int main()
{
    int n;
    while (1)
    {
        printf("1 과 10 사이의 정수를 입력해 주세요 : ");
        scanf("%d", &n);
        if (n > 0 && n <= 10) break;
        else printf("범위 오류!\n");
    }

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

<수 뒤집기>

```
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);

    int rev = 0;
    while (n > 0)
    {
        rev *= 10;
        rev += n % 10;
        n /= 10;
    }
    printf("%d", rev);
}
```

<십진수 이진수로 바꾸기>

```
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);

    int bi = 0;
    while (n > 0)
    {
        bi *= 10;
        bi += n % 2;
        n /= 2;
    }

    // 이진수가 거꾸로 저장되기 때문에 뒤집어줘야 한다.
    int rev = 0;
    while (bi > 0)
    {
        rev *= 10;
        rev += bi % 10;
        bi /= 10;
    }

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



```
for (char c = 'a'; c ≤ 'z'; c++);
```

```
for (int i = 1; i < n; i *= 2)
```

<소수 판별 알고리즘>

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

```
int main()
{
    int n, check = 1;
    scanf("%d", &n);
    for (int i = 2; i < n; i++)
    {
        if (n%i == 0)
        {
            check = 0;
            break;
        }
    }
    if (n <= 1) check = 0;
    if (check) printf("소수입니다.");
    else printf("소수가 아닙니다.");
}
```

<최대공약수 알고리즘>

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


```
int main()
{
    int a, b;
    scanf("%d%d", &a, &b);
    int c = (a > b ? b : a);
    int gcd;
    for (int i = c; i > 0; i--)
    {
        if (a%i == 0 && b%i == 0)
        {
            gcd = i;
            break;
        }
    }
    printf("%d", gcd);
}
```

풀어보세요!

- 수 뒤집기: 반복문을 통해 수를 뒤집어 봅시다

문제 풀이

- 2의 제곱인가?: for문의 성질을 이용해 간단히 풀 수 있습니다.
비트 연산을 통해 한 줄로 풀 수도 있습니다.



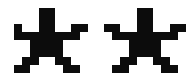
중첩 반복문

```
int main()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("i = %d  j = %d\n", i, j);
        }
    }
}
```

i	=	0	j	=	0
i	=	0	j	=	1
i	=	0	j	=	2
i	=	1	j	=	0
i	=	1	j	=	1
i	=	1	j	=	2
i	=	2	j	=	0
i	=	2	j	=	1
i	=	2	j	=	2

```
#include <stdio.h>

int main()
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j ≤ i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```



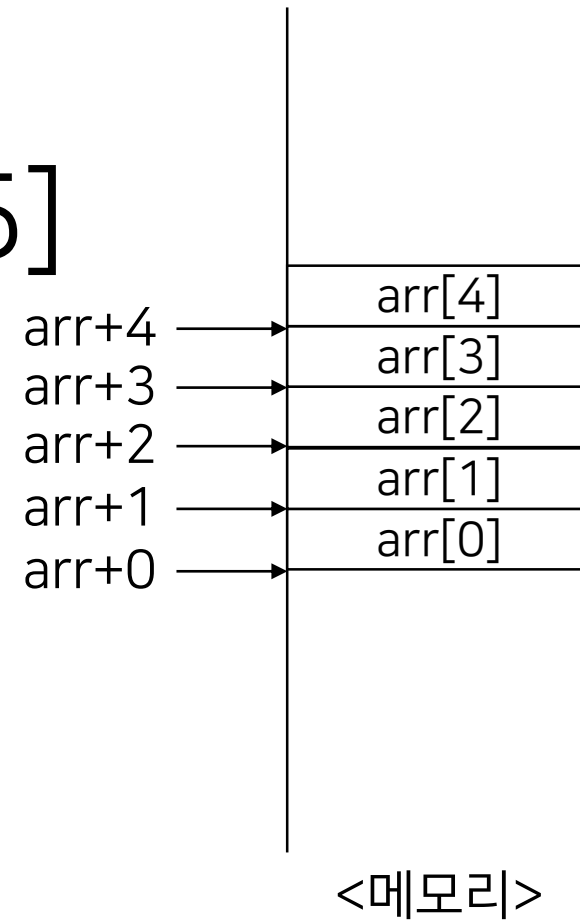
풀어보세요!

- [별찍기 2](#): 중첩 반복문을 통해 원하는 모양의 별 찍기
- [별찍기 5](#): 각 줄에서 무엇을 몇번 반복해서 출력할 지 잘 고민하세요



1차원 배열

int arr[5]



```
int arr[5] = {0};
```

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
0	0	0	0	0

```
for (int i = 0; i < 5; i++)  
{  
    scanf("%d", &arr[i]);  
}
```

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	2	3	4	5

← 1 2 3 4 5

<두 배열이 같은지 체크>

```
#include <stdio.h>

int main()
{
    int check = 1;
    int a[] = { 1,2,3,4,5 };
    int b[] = { 2,3,4,1,2 };

    for (int i = 0; i < 5; i++)
    {
        if (a[i] != b[i])
        {
            check = 0;
            break;
        }
    }
    if (check) printf("같다");
    else printf("다르다");
}
```

<배열이 오름차순인지 체크>

```
#include <stdio.h>

int main()
{
    int arr[5] = { 1,2,3,4,5 };

    int check = 1;
    for (int i = 0; i < 4; i++)
        if (arr[i] > arr[i + 1])
            check = 0;

    if (check) printf("오름차순");
    else printf("오름차순 아님");
}
```

<배열 정렬하기>

```
#include <stdio.h>

int main()
{
    int arr[5] = { 3,2,1,4,5 };

    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5 - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
}
```

풀어보세요!

- [음계](#) : 배열이 오름차순인지, 내림차순인지 판단하는 문제입니다.
다양한 방법으로 풀 수 있습니다.

문제 풀이

- [수 정렬하기](#) : 앞에 소개해드린 정렬 알고리즘을 적용해봅시다.



문자열 처리

- 변수의 이름은 알파벳, 숫자, _로 이루어 진다
- 변수의 이름은 숫자로 시작할 수 없고 C언어에 존재하는 키워드와 같은 이름으로는 만들 수 없다.
- 변수의 이름은 가급적 의미있게 짓는 것이 좋다.

정수형 자료형	크기	범위
char(character)	1바이트	-128 ~ 127
int(integer)	4바이트	-2,147,483,648 ~ 2,147,483,647
long long int	8바이트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807



수고하셨습니다!