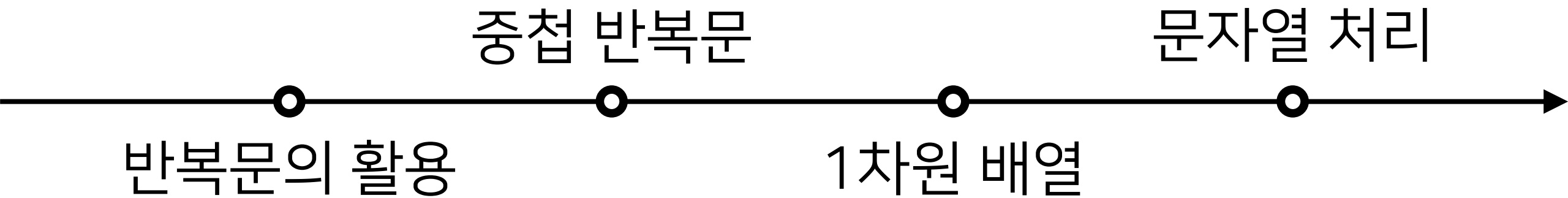


EC C언어 스터디

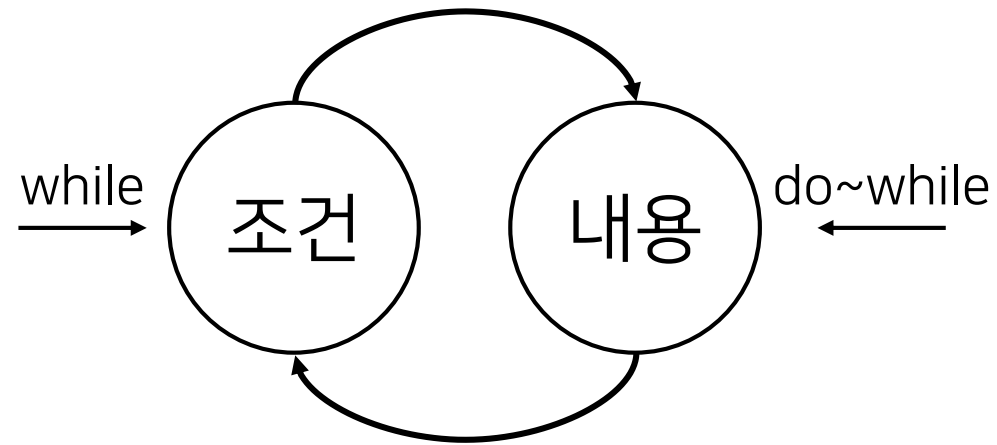
-3강-





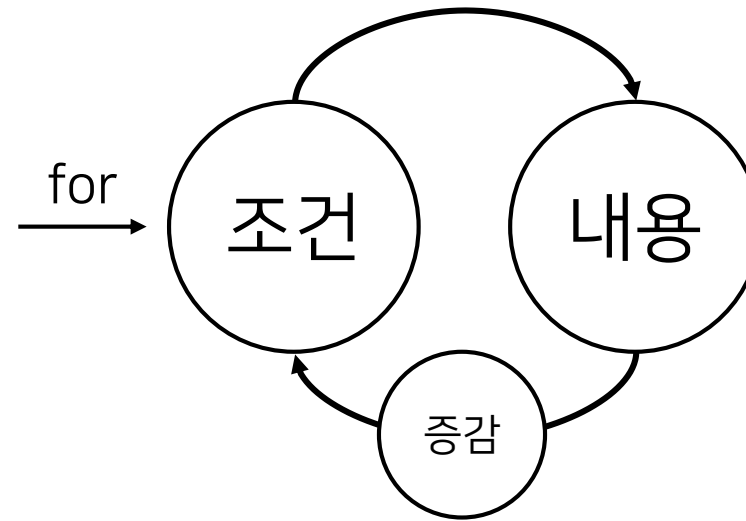
반복문의 활용

```
while(조건)  
{  
    내용  
}
```

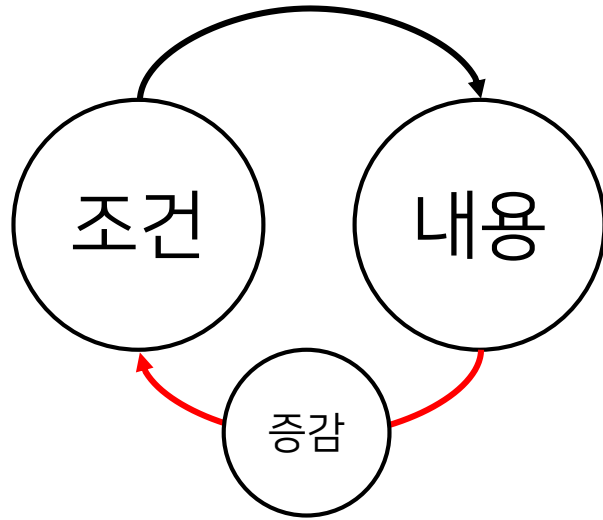


```
do  
{  
    내용  
} while(조건);
```

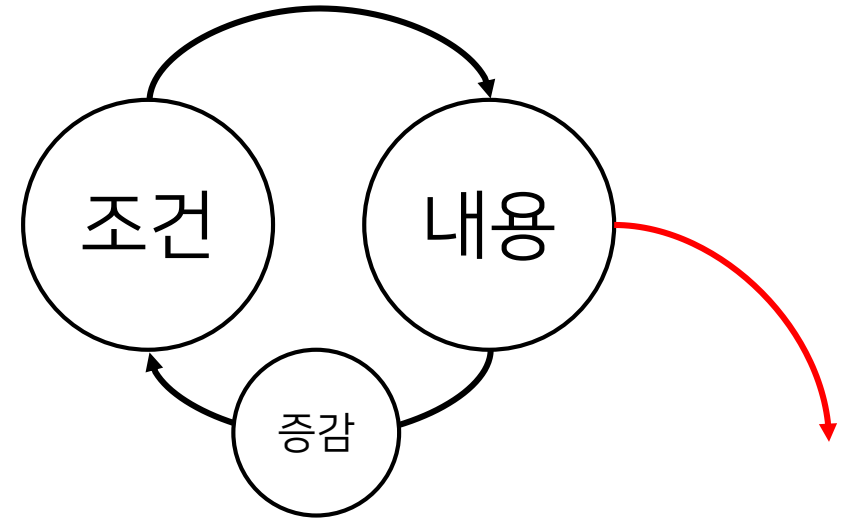
```
for(선언;조건;증감)  
{  
    내용  
}
```



- while문 : 단순 반복
- for문 : 반복 + 인덱싱



continue : 반복문의 나머지 내용
무시하고 다음으로 넘어감



continue : 반복문의 나머지 내용
무시하고 반복문을 종료함

<입력 값 유효성 검사하기>

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

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

```
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);
}
```


<숫자 한 자리씩 다루기>

```
int main()
```

```
{
```

```
    int n = 1234;
```

```
    while (n > 0)
```

```
    {
```

```
        printf("%d\n", n % 10);
```

```
        n /= 10;
```

```
    }
```

```
}
```

n=1234

n=123

n=12

n=1

n=0 -> 종료

n%10 = 4

n%10 = 3

n%10 = 2

n%10 = 1

n/=10

n/=10

n/=10

n/=10

————→ 4
 3
 2
 1

<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);
}
```

rev = 0

n = 1234

rev = 4

n = 123

rev = 43

n = 12

rev = 432

n = 1

rev = 4321

n = 0 -> 종료

<십진수 이진수로 바꾸기>

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

```
int main()
```

```
{
```

```
    int n;
```

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

```
    int bi[100];
```

```
    int len = 0;
```

```
    while (n > 0)
```

```
    {
```

```
        bi[len++] = n % 2;
```

```
        n /= 2;
```

```
    }
```

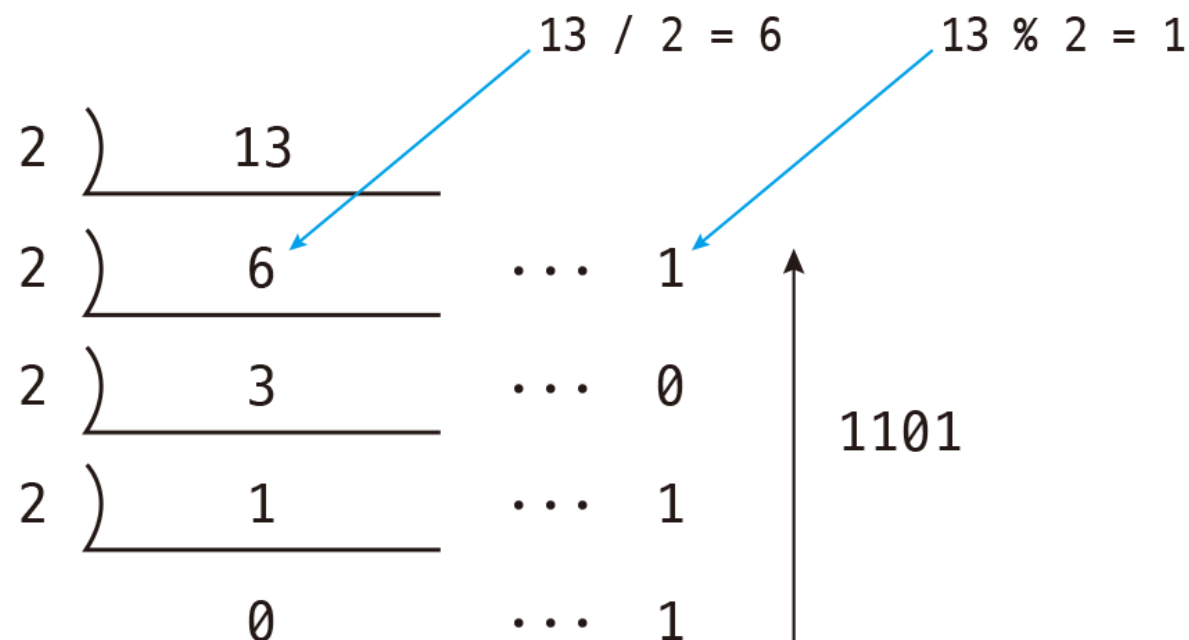
```
    for (int i = len - 1; i >= 0; i--)
```

```
    {
```

```
        printf("%d", bi[i]);
```

```
    }
```

```
}
```



```
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.
뒤집고 더하고 뒤집고...

문제 풀이

- 2의 제곱인가?: for문의 특성을 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!

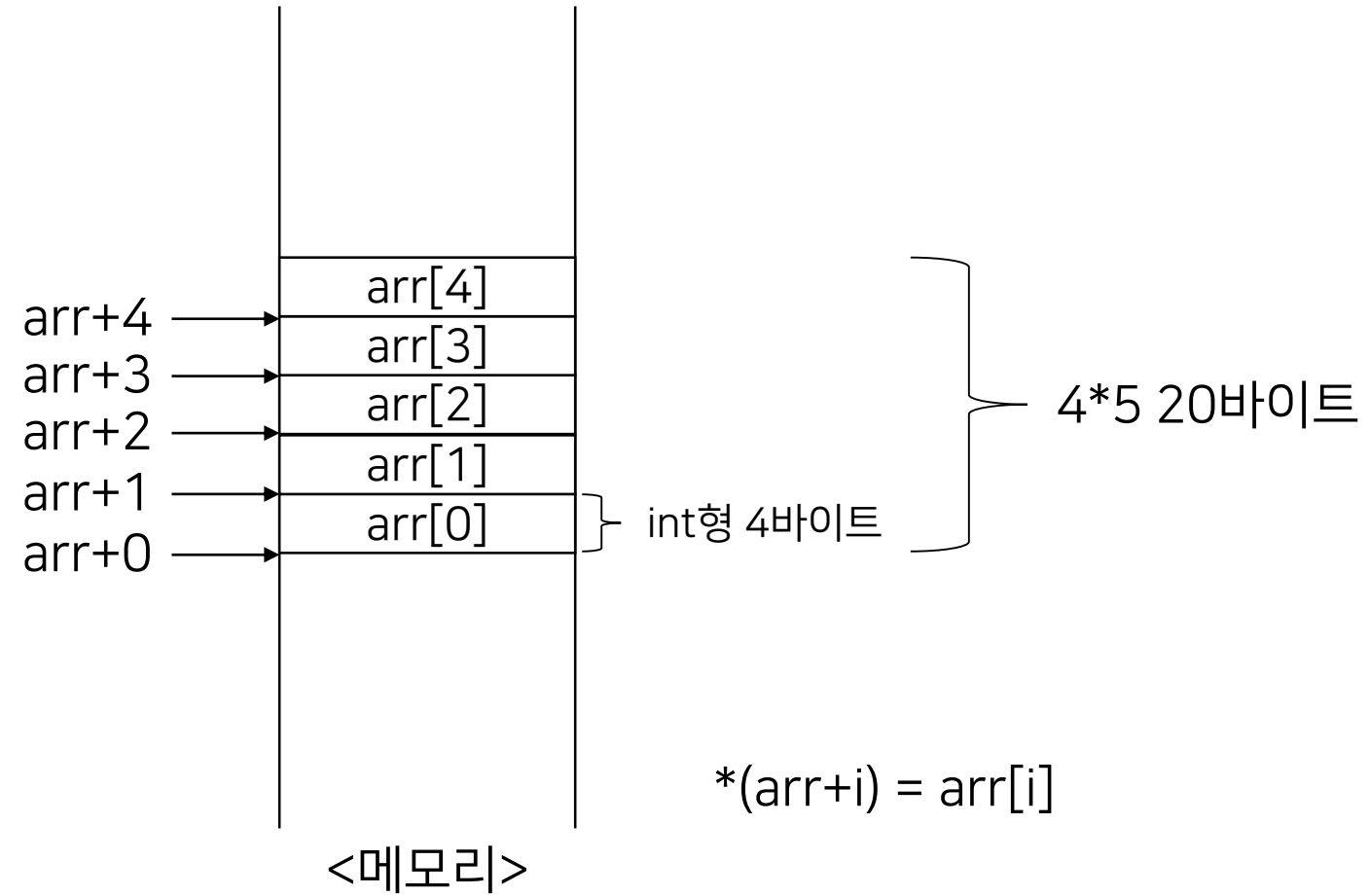
- [별찍기 2](#): 중첩 반복문을 통해 원하는 모양의 별 찍기
- [별찍기 5](#): 각 줄에서 무슨 문자를 몇번 반복해서 출력할 지 잘 고민하세요



1차원 배열

```
int num_1;  
int num_2;  
int num_3;  
int num_4;  
int num_5; scanf( "%d%d%d%d%d%d%d.....", &num_1, &num_2, &num_3);  
int num_6;  
int num_7; printf( "%d%d%d%d%d%d%d.....", num_1, num_2, sum_3);  
int num_8;  
~  
.  
.
```

```
int arr[5];
```



```
int arr[5];
```

선언할 때만 가능함



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

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

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


```
int arr[5] = { 1, 2 };
```

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

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

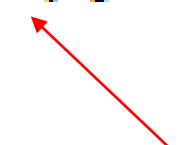
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
0	0	0	0	0

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

1 2 3 4 5
↑

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

```
int arr[5] = { 1,2,3,4,5 };  
  
for (int i = 1; i <= 5; i++)  
{  
    printf("%d ", arr[i]);  
}
```



i = 5 일때?

<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 check = 1;
    int arr[] = { 1,2,3,4,5 };

    for (int i = 0; i < 4; i++)
    {
        if (arr[i] > arr[i + 1])
        {
            check = 0;
            break;
        }
    }
    if (check) printf("오름차순");
    else printf("오름차순 아님");
}
```

<좌우 대칭 체크>

```
int main()
{
    int arr[7] = { 1,2,3,4,3,2,1 };

    int check = 1;
    for (int i = 0; i < 7 / 2; i++)
    {
        if (arr[i] != arr[7 - i - 1])
        {
            check = 0;
            break;
        }
    }
    if (check) printf("좌우 대칭!");
    else printf("좌우대칭 아님!");
}
```

<배열 정렬하기(버블 소트)>

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

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

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

모든 원소들끼리 비교한다

i = 0 -> j = 1 ~ 4

i = 1 -> j = 2 ~ 4

i = 2 -> j = 3 ~ 4

i = 3 -> j = 4

i = 4 -> j = X

풀어보세요!

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

문제 풀이

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



문자열 처리


- char를 일렬로 저장하겠다는 뜻.
- 1차원 배열과 똑같이 사용이 가능하며, 입출력과 문자열을 다룰 때 더 편하다.
- 문자열의 마지막에는 $\backslash 0$ (null)이 들어가야 한다


```
char s[5] = "abcd";
```

s[0]	s[1]	s[2]	s[3]	s[4]
'a'	'b'	'c'	'd'	'\0'

```
char a[100], b[100];  
scanf("%s%s", a, b);  
printf("a = %s, b = %s", a, b);
```

&를 붙이지 않는다




 Microsoft Visual Studio 디버그 콘솔

```
abc def  
a = abc, b = def
```


%[^\\n] = \\n을 제외하고 모두 받겠다

```
char a[100];  
scanf("%[^\\n]", a);  
printf("a = %s", a);
```

 Microsoft Visual Studio 디버그 콘솔

```
abc def  
a = abc def
```

```
char a[100];  
scanf("%s", a);  
printf("%s\n", a);  
printf("%s", a + 2);
```

 Microsoft Visual Studio 디버그 콘솔

```
abcde  
abcde  
cde
```

strlen(문자열)	문자열의 길이를 반환한다
strcpy(문자열1, 문자열2)	문자열 1에 문자열 2를 복사한다
strcmp(문자열1, 문자열2)	문자열을 비교한다(같으면 0)

<대문자 소문자로 바꾸기>

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

```
int main()
```

```
{
```

```
    char name[100];
```

```
    scanf("%[^\n]", name);
```

```
    int len = strlen(name);
```

```
    for (int i = 0; i < len; i++)
```

```
    {
```


```
        if (name[i] >= 'A' && name[i] <= 'Z')
```

```
            name[i] = name[i] - 'A' + 'a';
```

```
    }
```

```
    printf("%s", name);
```

```
}
```

 Microsoft Visual Studio 디버그 콘솔

Hwang In Seo
hwang in seo

'C' - 'A' = 2
2 + 'a' = 'c'

<문자열 좌우대칭 체크>

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

int main()
{
    char s[100] = "abcd dcba";

    int check = 1;
    int len = strlen(s);
    for (int i = 0; i < len / 2; i++)
    {
        if (s[i] != s[len - i - 1])
        {
            check = 0;
            break;
        }
    }
    if (check) printf("좌우 대칭!");
    else printf("좌우대칭 아님!");
}
```


풀어보세요!

- [문자열 반복](#) : 문자열과 중첩 반복문
- [그대로 출력하기](#) : %[^\n]를 이용해 보세요

문제 풀이

- [수들의 합](#) : 숫자형으로는 100자리가 넘는 숫자를 입력받을 수 없습니다.
문자열로 입력을 받은 뒤 처리하는 것이 좋습니다.
- [단어 공부](#) : 문자열과 배열을 동시에 사용하는 문제입니다.



수고하셨습니다!