

## (훈련반1) Level20.5

이곳에서는 for문 응용문제들과 복습문제, 재귀호출 문제들을 풀게 됩니다.

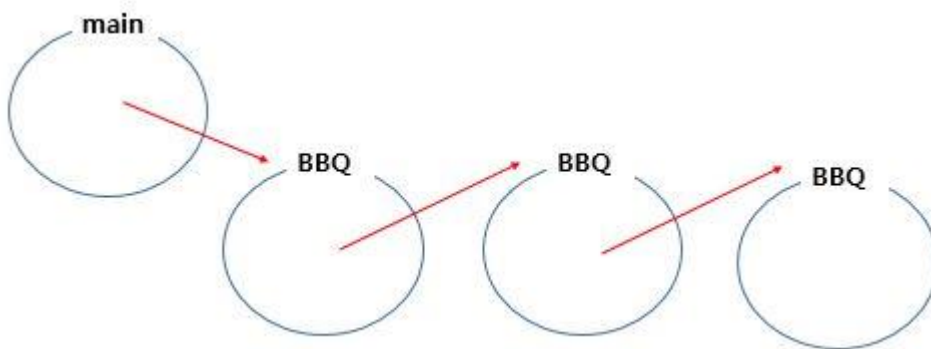
for문 응용문제는 Level20을 기점으로, 다양한 문제들이 출제됩니다.

어떻게 짜야할지 최대한 세부설계를 한 뒤, 코딩을 시작 해 보세요

## Level20.5 BBQ 갔다오기 [난이도 : 3]

문제 1번 [\[숙제 목록보기\]](#)

그림과 같이 동작하는 프로그램을 만들어주세요. (입출력 값 없음)



```
#include <iostream>
using namespace std;

void BBQ(int num)
{
    if (num == 3)
        return;
    BBQ(num + 1);
}

int main(void)
{
    int num = 1;

    BBQ(num);

    return 0;
}
```

# Level20.5 앞으로 돌진하는 계단

## [난이도 : 2]

문제 2번 [[숙제](#) [목록보기](#)]

한 문장을 입력 받으세요.(최대 10글자)

그리고 예제와 같이 계단식으로 출력하세요.

중첩 2중 for문을 이용해서 풀어주세요

for문을 돌리는 방향 및 순서를 유의 해 주세요

## 입력 예제

78ATQP

## 출력 결과

P

QP

TQP

ATQP

8ATQP

78ATQP

```
#include <iostream>
using namespace std;

int main(void)
{
    char str[10] = {};
    cin >> str;

    bool isEnd = false;
    int len = 0;
    while (!isEnd)
    {
        if (str[len] == '\0')
            isEnd = true;
        else
            len++;
    }

    for (int i = 0; i < len; ++i)
    {
        int startIdx = len - 1 - i;
        for (int k = startIdx; k < len; ++k)
        {
            cout << str[k];
        }
        cout << endl;
    }

    return 0;
}
```

# Level20.5 절반 나누기 [난이도 : 2]

문제 3번 [[숙제](#) [목록보기](#)]

한 문장을 입력 받으세요.(최대 10글자)

그리고 문장의 길이/2 를 하여 2등분 해주세요.

예를들어 "ABCDEAB" 문장을 2등분하면  $7/2=3$  이므로

0~2번 index 글자: ABC

3~6번 index 글자: DEAB

이렇게 2등분 할 수 있습니다.

이렇게 나누어진 두 문장이 동일한 문장인지 확인하는 프로그램을 작성 해주세요.

ex1)

입력: ABCABC

출력: 동일한문장

ex2)

입력: ABCDEAB

출력: 다른문장

## 입력 예제

ABCABC

## 출력 결과

동일한문장

```
#include <iostream>
using namespace std;

int main(void)
{
    char input[20] = {};
    cin >> input;

    bool isEnd = false;
    int len = 0;
    while (!isEnd)
    {
        if (input[len] == '\\0')
            isEnd = true;
        else
            len++;
    }

    bool isSame = false;
    if (len % 2 != 1)
    {
        int div = len / 2;
        int cnt = 0;
        for (int i = 0; i < div; ++i)
        {
            if (input[i] == input[div + i])
                cnt++;
        }
        if (cnt == div)
            isSame = true;
    }

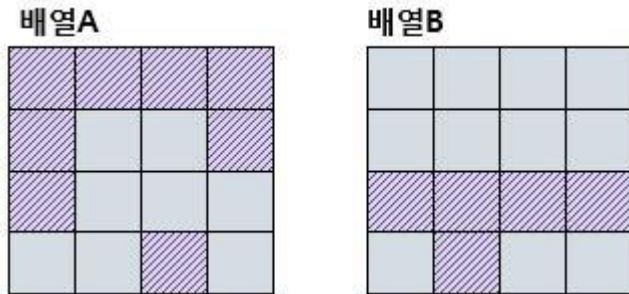
    if (isSame)
        cout << "동일한문장";
    else
        cout << "다른문장";

    return 0;
}
```

# Level20.5 겹치지 않도록 [난이도 : 3]

문제 4번 [숙제 [목록보기](#)]

두 비트배열을 입력 받아주세요.



두 비트배열을 겹쳤을때,

색칠한 부분이 겹치면, "걸리다" 출력

겹치는 부분이 없다면, "걸리지않는다" 라고 출력 해주세요.

(색칠된곳은 1 로 입력을 받고, 색칠안된곳은 0 으로 입력 받으면 됩니다)

## 입력 예제

```
1 1 1 1
1 0 0 1
1 0 0 0
0 0 1 0
```

```
0 0 0 0
0 0 0 0
1 1 1 1
0 1 0 0
```

## 출력 결과

걸리다

```
#include <iostream>
using namespace std;

int main(void)
{
    bool arrA[4][4] = {};
    bool arrB[4][4] = {};
    for (int i = 0; i < 16; ++i)
        cin >> arrA[i / 4][i % 4];
    for (int k = 0; k < 16; ++k)
        cin >> arrB[k / 4][k % 4];

    bool arrA2[16] = {};
    bool arrB2[16] = {};
    for (int i = 0; i < 16; ++i)
        arrA2[i] = arrA[i / 4][i % 4];
    for (int k = 0; k < 16; ++k)
        arrB2[k] = arrB[k / 4][k % 4];

    bool isDouble = false;
    for (int o = 0; o < 16; ++o)
    {
        if (arrA2[o] && arrB2[o])
        {
            isDouble = true;
            break;
        }
    }
    if (isDouble)
        cout << "걸리다";
    else
        cout << "걸리지 않는다.";

    return 0;
}
```

# Level20.5 일곱계단 만들기 [난이도 : 4]

문제 5번 [[숙제 목록보기](#)]

문자 1개를 입력받으세요

ASCII코드값 기준 입력 받은 문자에서 부터 -3 문자부터 +3 문자까지 출력하면 됩니다.

만약 G를 입력받았다면 DEF**G**HIJ 를 출력하면 됩니다.

만약 N를 입력받았다면 KLM**N**OPQ 를 출력하면 됩니다.

그런데 만약 출력해야 하는 문자가 A 이전이라면, Z를 출력하면 되고,

만약 출력해야 하는 문자가 Z 다음 문자라면 A를 출력하면 됩니다.

따라서

만약 Y를 입력받았다면 VWX**Y**ZAB 를 출력하면 되고,

만약 B를 입력받았다면 YZAB**B**CDE를 출력하면 됩니다.

## 입력 예제

Y

## 출력 결과

VWXYZAB



```

#include <iostream>
using namespace std;

int main(void)
{
    char input = '\0';
    cin >> input;

    // ascii -> A : 65 , Z : 90

    char arr[7] = {};
    arr[3] = input;
    int idx = 2;
    for (int i = 0; i < 3; ++i)
    {
        char val = input - (i+1);
        if (val < 65)
        {
            // 64 -(+26)-> 90
            val += 26;
        }
        arr[idx - i] = val;
    }
    idx = 4;
    for (int i = 0; i < 3; ++i)
    {
        char val = input + (i + 1);
        if (val > 90)
        {
            // 91 -(-26)-> 65
            val -= 26;
        }
        arr[idx + i] = val;
    }

    for (int i = 0; i < 7; ++i)
    {
        cout << arr[i];
    }

    return 0;
}

```

# Level20.5 성쌓기 [난이도 : 2]

문제 6번 [[숙제](#) [목록보기](#)]

숫자 7개를 배열에 입력 받아주세요.

1층 3번 index까지 출력

2층 4번 index까지 출력

3층 5번 index까지 출력

4층 6번 index까지 출력

ex)

3 5 7 1 4 2 8 을 입력 받았다면 아래와 같이 출력 하면 됩니다.

3 5 7 1

3 5 7 1 4

3 5 7 1 4 2

3 5 7 1 4 2 8

중첩 2중 for문을 활용해서 출력 해주세요.

## 입력 예제

3 5 7 1 4 2 8

## 출력 결과

3 5 7 1

3 5 7 1 4

3 5 7 1 4 2

3 5 7 1 4 2 8

```
#include <iostream>
using namespace std;

int main(void)
{
    int input[7] = {};
    for (int i = 0; i < 7; ++i)
        cin >> input[i];
    for (int layer = 0; layer < 4; ++layer)
    {
        int lastIdx = layer + 4;
        for (int idx = 0; idx < lastIdx; ++idx)
        {
            cout << input[idx] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

# Level20.5 늘어나는 글자 [난이도 : 2]

문제 7번 [[숙제](#) [목록보기](#)]

한문장을 입력 받으세요.

한문장을 아래와 같이 출력 해주세요.

ex)

[입력]

BBQWORLD

[출력]

B

BB

BBQ

BBQW

BBQWO

BBQWOR

BBQWORL

BBQWORLD

ex)

[입력]

GDPK

[출력]

G

GD

GDP

GDPK

# 입력 예제

BBQWORLD

# 출력 결과

B

BB

BBQ

BBQW

BBQWO

BBQWOR

BBQWORLD

BBQWORLD

```
#include <iostream>
using namespace std;

int main(void)
{
    char input[20] = {};
    cin >> input;

    bool isEnd = false;
    int len = 0;
    while (!isEnd)
    {
        if (input[len] == '\\0')
            isEnd = true;
        else
            len++;
    }
    for (int layer = 0; layer < len; ++layer)
    {
        int lastIdx = layer + 1;
        for (int idx = 0; idx < lastIdx; ++idx)
        {
            cout << input[idx];
        }
        cout << endl;
    }

    return 0;
}
```

# Level20.5 두배열 머지(Merge)하기

[난이도 : 6]

문제 8번 [숙제 목록보기]

정렬되어 있는 네칸짜리 두 배열에 각각 숫자 4개씩 입력 받습니다.

두배열에 있는 8개 숫자들을 합쳐 하나의 배열에 정렬된 상태로 넣으려고 합니다.

아래의 알고리즘으로 동작되도록 코딩해주세요.

ex) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2	3	3	3	4	6	7
---	---	---	---	---	---	---	---

두개의 배열을 합쳐 한 개의 Result 배열에 채우려 합니다.

과정1) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

--	--	--	--

가장 첫번째 숫자중 작은 숫자를 Result에 넣고 화살표를 한칸 옮깁니다.

과정2) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1			
---	--	--	--

과정3) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2		
---	---	--	--

같은 숫자라면 아무 숫자나 선택해도 됩니다.

과정4) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2	3	
---	---	---	--

과정5) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2	3	3
---	---	---	---

과정6) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2	3	3	3	
---	---	---	---	---	--

과정7) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

 → Result 

1	2	3	3	3	4	
---	---	---	---	---	---	--

과정8) 

1	3	3	7
---	---	---	---

2	3	4	6
---	---	---	---

끝
---

 → Result 

1	2	3	3	3	4	6
---	---	---	---	---	---	---

한쪽이 끝났기에 반대쪽 숫자들을 모두 등록해주면 Result 

1	2	3	3	3	4	6	7
---	---	---	---	---	---	---	---

## 입력 예제

1 3 3 7

2 3 4 6

## 출력 결과

1 2 3 3 3 4 6 7



```

#include <iostream>
using namespace std;

// 인자값으로 들어온 배열을 오름차순으로 정렬해주는 함수
void func(int* arr)
{
    int table[10] = {};
    for (int i = 0; i < 4; ++i)
    {
        int idx = arr[i];
        table[idx]++;
    }

    int cnt = 0;
    int pos = 0;
    while (cnt < 4)
    {
        if (table[pos] != 0)
        {
            for (int i = 0; i < table[pos]; ++i)
            {
                arr[cnt] = pos;
                cnt++;
            }
            pos++;
        }
    }
}

int main(void)
{
    int input1[4] = {};
    int input2[4] = {};
    // 값 입력받기
    for (int i = 0; i < 8; ++i)
    {
        if (i < 4)
            cin >> input1[i];
        else
        {
            int i2 = i % 4;
            cin >> input2[i2];
        }
    }
}

```

```

int sum[8] = {}; // 합쳐진 값이 들어갈 배열
func(input1);
func(input2);

int idx1 = 0;
int idx2 = 0;
for (int i = 0; i < 8; ++i)
{
    if (idx1 < 4 && idx2 < 4)
    {
        if (input1[idx1] < input2[idx2])
        {
            sum[i] = input1[idx1];
            idx1++;
        }
        else
        {
            sum[i] = input2[idx2];
            idx2++;
        }
    }
    else if (idx1 < 4)
    {
        sum[i] = input1[idx1];
        idx1++;
    }
    else
    {
        sum[i] = input2[idx2];
        idx2++;
    }
}

for (int i = 0; i < 8; ++i)
    cout << sum[i] << " ";

return 0;
}

```

# Level20.5 원하는 패턴의 크기 적용

[난이도 : 3]

문제 9번 [[숙제](#) [목록보기](#)]

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

위 배열을 하드코딩 하주세요. 패턴 size를 입력 받으세요.

만약 2,2 를 입력 받았다면


2x2 패턴을 적용 시키면 됩니다.

그리고 패턴을 적용시키면 합을 구할 수 있습니다.

ex) 2,2 size의 패턴을 0,0에 적용시키면 합은  $3+5+3+3=14$  입니다.

패턴의 size를 입력받고 패턴을 적용시켜 합을 구하여. max값이 나오는 위치를 출력 하주세요.

주의 : 입력하실때 y좌표부터 입력받아주세요 ex) 2,3은 y=2, x=3

## 입력 예제

2 2

## 출력 결과

(2,3)

```

#include <iostream>
using namespace std;

int main(void)
{
    int arr2D[4][5] =
    {
        3,5,4,2,5,
        3,3,3,2,1,
        3,2,6,7,8,
        9,1,1,3,2
    };

    int sizeX = 0, sizeY = 0;
    cin >> sizeX >> sizeY;

    int maxSum = -100;
    int maxY = -1;
    int maxX = -1;
    for (int y = 0; y < 4; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            int chY = y + sizeY;
            int chX = x + sizeX;
            if (chY >= 0 && chY <= 4 && chX >= 0 && chX <= 5) // 조건 주의하기
            {
                int sum = 0;
                for (int y2 = 0; y2 < chY; ++y2)
                {
                    for (int x2 = 0; x2 < chX; ++x2)
                    {
                        sum += arr2D[y2][x2];
                    }
                }
                if (sum > maxSum)
                {
                    maxSum = sum;
                    maxY = y;
                    maxX = x;
                }
            }
        }
    }

    cout << "(" << maxY << ", " << maxX << ")";

    return 0;
}

```