

## (훈련반2) Level30.5

훈련반의 마지막 문제입니다. 고난이도의 for문 응용문제들 풀어봅시다.

항상 습관처럼 설계를 미리 꼼꼼히 한 후 (계획을 구체적으로 세운 후), 문제 푸는 것을 잊지마세요.

## Level30.5 원형시계 돌리기 (난이도 : ★★☆☆)

문제 1번 [숙제 목록보기]

원형시계가 있습니다.

몇도 돌릴것인지 90도의 배수로 입력 받아주세요.

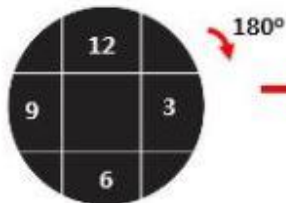
그리고 시계를 시계방향으로 돌리고, 그 결과를 출력 하세요.



원형 시계

Ex)

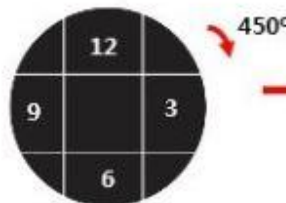
회전각도 입력: 180



출력 결과 : 6, 3, 9, 12



회전각도 입력: 450



출력 결과 : 9, 6, 12, 3



## 입력 예제

180

## 출력 결과

6 3 9 12

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

const int timeCnt = 4;
int times[timeCnt] = {12, 3, 6, 9};

void rotate(int _angle)
{
    int rotateCnt = _angle / 90;
    int temp[timeCnt] = {};
    for (int i = 0; i < timeCnt; ++i)
    {
        int idx = (i + rotateCnt) % 4;
        temp[idx] = times[i];
    }
    memcpy(times, temp, sizeof(int) * timeCnt);
}

void printTime()
{
    cout << times[0] << " ";
    cout << times[3] << " ";
    cout << times[1] << " ";
    cout << times[2] << " ";
};

int main(void)
{
    int angle = 0;
    cin >> angle;
    rotate(angle);
    printTime();

    return 0;
}
```

# Level30.5 승부차기

문제 2번 [숙제 목록보기]

축구 결승전에서 승부차기를 하게 되었습니다.

0x0x 이면 첫번째/세번째 사람이 공을 넣은 것입니다.

승부차기 할 **사람수를 입력** 받고, **가능한 경우수를 출력** 하세요.

(재귀호출로 구현해주세요)

ex)

[입력]

3

[출력]

000

00X

0X0

0XX

X00

X0X

XX0

XXX

## 입력 예제

4

## 출력 결과

0000

000X

00X0

00XX

0X00

0X0X

0XX0

0XXX

X000

X00X

X0X0

X0XX

XX00

XX0X

XXX0

XXXX

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

int people = 0;
const int goalCase = 2;
int* resultArr = nullptr;

void init()
{
    cin >> people;
    resultArr = new int[people];
    for (int i = 0; i < people; ++i)
        resultArr[i] = 0;
}

void print()
{
    for (int k = 0; k < people; ++k)
    {
        if (resultArr[k] == 0)
            cout << 'o';
        else if (resultArr[k] == 1)
            cout << 'x';
        else
            __noop;
    }
    cout << endl;
}

void dfs(int _level)
{
    if (_level == people)
    {
        print();
        return;
    }
    for (int i = 0; i < goalCase; ++i)
    {
        resultArr[_level] = i;
        dfs(_level + 1);
        resultArr[_level] = '\0';
    }
}
```

```
void proc()
{
    int initLevel = 0;
    dfs(initLevel);
}

void end()
{
    delete[] resultArr;
}

int main(void)
{
    init();
    proc();
    end();

    return 0;
}
```

# Level30.5 n번째로 큰숫자 출력

## (난이도 : ★☆☆)

문제 3번 [[숙제](#) [목록보기](#)]

아래 배열을 하드 코딩 해주세요.

1	5	4	2	-5	-7
---	---	---	---	----	----

n을 입력받고, n번째 큰 숫자를 출력 해주세요.

만약 3을 입력받았다면, 세 번째로 큰 숫자는  $5 > 4 > 2$ 이기 때문에 2를 출력하면 됩니다.

## 입력 예제

3

## 출력 결과

2

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

const int length = 6;
int arr[length] = { 1,5,4,2,-5,-7 };
int n = 0;

void selectSort()
{
    for (int i = 0; i < length-1; ++i)
    {
        for (int k = i + 1; k < length; ++k)
        {
            if (arr[i] < arr[k])
            {
                int temp = arr[k];
                arr[k] = arr[i];
                arr[i] = temp;
            }
        }
    }
}

void nMax()
{
    int max = arr[n - 1];
    cout << max;
}

void proc()
{
    selectSort();
    nMax();
}

int main(void)
{
    cin >> n;
    proc();

    return 0;
}
```



# Level30.5 2진수로 된 가장 큰수를 찾아라

## (난이도 : ★★★)

문제 4번 [[숙제](#) [목록보기](#)]

0과 1로 된 문자열 3개 입력 받으세요. (최대 30글자)

2진수라고 생각 했을 때, 가장 큰 숫자를 출력 해주세요.

ex)

[입력]

1101011

11110

1101110

[출력]

1101110

## 입력 예제

1001011

11110

1101110

## 출력 결과

1101110

```

#include <iostream>
using namespace std;

const int maxLength = 30;
const int inputCnt = 3;
char inputs[inputCnt][maxLength + 1] = {};
int values[inputCnt] = {};
const int one = 49;      // ascii -> '0' = 48, '1' = 49

void init()
{
    for (int i = 0; i < inputCnt; ++i)
        cin >> inputs[i];
}

void binToDecim()
{
    for (int k = 0; k < inputCnt; ++k)
    {
        int len = strlen(inputs[k]);
        int sum = 0;
        int weight = 1;
        for (int o = len-1; o >= 0; --o)
        {
            if (inputs[k][o] == one)
            {
                sum += weight;
            }
            weight *= 2;
        }
        values[k] = sum;
    }
}

int findMax(int* _arr, int _len)
{
    int maxIdx = 0;
    int max = _arr[maxIdx];
    for (int i = 1; i < _len; ++i)
    {
        if (max < _arr[i])
        {
            max = _arr[i];
            maxIdx = i;
        }
    }
    return maxIdx;
}

```

```
void print(int _idx)
{
    cout << inputs[_idx];
}

void proc()
{
    binToDecim();
    int maxIdx = findMax(values, inputCnt);
    print(maxIdx);
}

/*
    input
    1001011 = 75
    11110   = 30
    1101110 = 110
*/

int main(void)
{
    init();
    proc();

    return 0;
}
```

# Level30.5 후보 선출하기

문제 5번 [숙제 목록보기]

후보들을 입력 받으세요.

그리고 입력 한 후보 중 몇명을 선출할지 뽑을 인원을 입력 받으세요.(숫자 1개 입력)

중복을 허용하여 뽑을 수 있는 후보의 경우를 출력 해주세요.

(Backtracking을 이용해서 풀어주세요)

Ex)

후보입력 : AGFT ↴

뽑을 인원 : 2 ↴

<출력>

AA

AG

AF

AT

GA

GG

GF

GT

...

TF

TT

## 입력 예제

AGFT

2

## 출력 결과

AA

AG

AF

AT

GA

GG

GF

GT

FA

FG

FF

FT

TA

TG

TF

TT

```

#include <iostream>
using namespace std;

char candidates[20] = {};
int select = 0;
char* selectCase = nullptr;

void dfs(int _level)
{
    if (_level == select)
    {
        cout << selectCase << endl;
        return;
    }

    for (int i = 0; i < strlen(candidates); ++i)
    {
        selectCase[_level] = candidates[i];
        dfs(_level + 1);
        selectCase[_level] = '\0';
    }
}

int main(void)
{
    cin >> candidates;
    cin >> select;

    selectCase = new char[select + 1];
    for (int i = 0; i < select + 1; ++i)
        selectCase[i] = '\0';
    int initLevel = 0;
    dfs(initLevel);

    delete[] selectCase;
    return 0;
}

```

# Level30.5 자전거열쇠 비밀번호 맞추기

문제 6번 [숙제 목록보기]



A ~ Z 까지 대입을 해봐서 암호를 알아내려고 합니다.

아래의 순서대로 대입을 해보려합니다.

AAAA

AAAB

AAAC

...

ZZZZ

자전거 암호 n개가 존재할때, 몇회만에 각 암호를 해제 할 수 있는지 출력하세요.

ex)

[입력]

3 (자전거 암호 개수)

AAAC (암호 입력)

ATKC (암호 입력)

ZBAB (암호 입력)

[출력]

3

13107

440078

## 입력 예제

3

AAAC

ATKC

ZBAB

## 출력 결과

3

13107

440078



```

#include <iostream>
using namespace std;

int pwCnt = 0; // 암호 개수
const int pwLength = 4; // 암호 자리수
char (*pw)[pwLength + 1] = nullptr; // 암호 저장
char temp[pwLength + 1] = {}; // 암호 임시저장
int alphaCnt = 'Z' - 'A' + 1; // 26개
int offset = 65; // ascii -> 'A' = 65 , 'Z' = 90
const int same = 0;
int cnt = 0;
bool isFind = false;

void dfs(int _level, int _pwNum)
{
    if (_level == pwLength)
    {
        ++cnt;
        if (strcmp(temp, pw[_pwNum]) == same) // 같으면 0을 반환
        {
            isFind = true;
            cout << cnt << endl;
        }
        return;
    }
    for (int i = 0; i < alphaCnt; ++i)
    {
        if (isFind)
            break;
        temp[_level] = i + offset;
        dfs(_level + 1, _pwNum);
        temp[_level] = '\0';
    }
}

int main(void)
{
    cin >> pwCnt;
    pw = new char[pwCnt][pwLength + 1];
    for (int i = 0; i < pwCnt; ++i)
        cin >> pw[i];

    for (int i = 0; i < pwCnt; ++i)
    {
        cnt = 0;
        isFind = false;
        dfs(0, i);
    }
}

```

```
    }  
  
    delete[] pw;  
    return 0;  
}
```

# Level30.5 다섯형제 1020

문제 7번 [숙제 목록보기]

다섯 숫자를 아래 배열에 입력 받으세요.

--	--	--	--	--

입력받은 숫자들 중 몇 개를 뽑아서

더했을 때 값이 10~20 되는 조합이 총 몇 가지 인지 Counting 해주세요

(Backtracking으로 구현 해 주세요)

ex)

[입력]

1 3 4 7 9

[출력]

3+4+7

3+4+9

4+7

7+9

1+3+4+7

...

-----  
18

[힌트]

이진트리 형태로 생각해보면 됩니다.

입력이 1 3 5 7 9 라면

첫번째 Level에서는 숫자 1을 쓸지 or 말지

두번째 Level에서는 숫자 3을 쓸지 or 말지

세번째 Level에서는 숫자 5를 쓸지 or 말지

네번째 Level에서는 숫자 7를 쓸지 or 말지

다섯번째 Level에서는 숫자 9를 쓸지 or 말지

## 입력 예제

5 4 3 9 1

## 출력 결과

16

```

#include <iostream>
using namespace std;

const int length = 5;
int arr[length] = {};
int isUse[length] = {};
int sum = 0;
int cnt = 0;

void check()
{
    for (int i = 0; i < length; ++i)
    {
        if (isUse[i] == 1)
            sum += arr[i];
    }
    if (sum >= 10 && sum <= 20)
        ++cnt;
    sum = 0;
}

void dfs(int _level)
{
    if (_level == length)
    {
        check();
        return;
    }

    for (int i = 0; i < 2; ++i)
    {
        isUse[_level] = i;
        dfs(_level + 1);
        isUse[_level] = 0;
    }
}

int main(void)
{
    for (int i = 0; i < length; ++i)
        cin >> arr[i];

    dfs(0);
    cout << cnt;

    return 0;
}

```

# Level30.5 다시 풀어보는 출동 순서

## (난이도 : ★★★)

문제 8번 [숙제 목록보기]

B, I, A, H 슈퍼영웅들 중 출동할 사람을 순서대로 뽑아야 합니다.

척척박사님은 자신의 이름이 다섯글자이기에 영웅B를 시작으로 n번째 사람을 선택합니다.


출동하는 영웅들의 순서를 출력 하세요.

(큐를 이용하지 않고, For문 or While문을 활용해서 풀어주세요)


만약 5를 입력받았다면,

항상 다섯번째 사람을 먼저 출동시키면 됩니다.

**B**  


**I**  


**A**  


**H**  


**B**  
①  
②  
③  
④  
**5**

→ 첫번째 출동자는 배동맨

~~**B**~~  
①  
④  
**5**

→ 두번째 출동자는 캡틴A

~~**B**~~  
~~**I**~~  
~~**A**~~  
**H**  
①  
③  
④  
**5**

→ 세번째 출동자는 헐크으

출동순서 결과: B A H I

## 입력 예제

5

## 출력 결과

B A H I

```

#include <iostream>
using namespace std;

int main(void)
{
    const int heroCnt = 4;
    char heros[heroCnt + 1] = "BIAH";
    int heroIdx = -1;           // 현재 선택된 히어로의 인덱스
    int loopCnt = 0;           // 몇번째 반복인지 나타낸다.
    int selectCnt = 0;         // 몇명이 뽑혔는지
    int isSelect[heroCnt] = {}; // 선택됐는지
    int maxLoop = 0;

    cin >> maxLoop;

    while (true)
    {
        heroIdx = (++heroIdx) % heroCnt;
        if (isSelect[heroIdx] == 1)
            continue;
        ++loopCnt;
        if (loopCnt % maxLoop == 0)
        {
            cout << heros[heroIdx] << " ";
            isSelect[heroIdx] = 1;
            ++selectCnt;
        }
        if (selectCnt == heroCnt)
            break;
    }

    return 0;
}

```