Level23 1등, 2등, 3등 선물주기

[난이도 : 3]

문제 1번 [숙제 목록보기]

네 명의 친구들의 이름을 입력 받아 주세요. (사람당 한글자씩, 총 4글자)

그리고 이 친구들 중 1등, 2등, 3등을 뽑아 선물을 주려고 합니다.

한 사람은 하나의 선물만 받을 수 있습니다.

선물을 줄 수 있는 경우를 모두 출력 해주세요.

ex)

[입력]

ATKP

[출력]

ATK

ATP

AKT

AKP

APT

APK

TAK

TAP

TKA

TKP

TPA

TPK

KAT

KAP

KTA

KTP

KPA

KPT

PAT

PAK

PTA

PTK

PKA

PKT

[TIP : 중복순열과 순열의 차이]

A,B,C,D 중 2장을 뽑는 경우의 수

중복순열 : 16가지 (AA, AB, AC, AD / BA, BB, BC, BD / CA, CB, CC, CD / DA, DB, DC, DB)

순열 : 4가지 (AB, AC, AD / BA, BC, BD / CA, CB, CD / DA, DB, DC)

입력 예제

`ATKP`

출력 결과

`ATK`

`ATP`

`AKT`

`AKP`

`APT`

APK

TAK

TAP

TKA

TKP

TPA

TPK

KAT

KAP

KTA

KTP

KPA

KPT

PAT

PAK

PTA

PTK

PKA

PKT

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

char names[5] = {};                    // 이름 저장하는 배열(4명)
const int maxRank = 3;           // 순위 범위
const int people = 4;            // 사람 수
char ranker[3] = {};
int ranked[4] = {};
int cnt = 0;

void recursive(int rank)
{
        if (rank == maxRank)
        {
                cout << ranker << endl;
                cnt++;
                return;
        }

        for (int i = 0; i < people; ++i)
        {
                if (ranked[i] == 0)
                {
                        ranker[rank] = names[i];
                        ranked[i] = 1;
                        recursive(rank + 1);
                        ranker[rank] = '\0';
                        ranked[i] = 0;
                }
        }
}

int main(void)
{
        cin >> names;

        int initRank = 0;
        recursive(initRank);
        cout << "total case : " << cnt;
        return 0;
}
```

Level23 다툰친구 B와 T [난이도 : 4]

문제 2번 [숙제 목록보기]

네 글자를 입력 받으세요.

네 글자를 조합하여 나올수 있는 모든 경우가 몇가지인지 알아내고자 합니다.

그런데 **B와 T 글자는 서로 붙어있으면 안됩니다.**

재귀호출을 이용해서 풀어주세요

ex)

만약, BOTK 네글자를 입력받았다면,

BBBB -> 가능

BBBT -> 불가능

BOOT -> 가능

TTBK -> 불가능

TTTK -> 가능

네 글자를 입력받고,

B와 T글자가 서로 붙어있지 않은 총 경우의 수가 몇 가지인지 출력하세요.

입력 예제

```
BOTT
```

출력 결과

```
120
```

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

char inputs[5] = {};              // 글자 저장하는 배열
const int length = 4;             // 글자 길이
const int charCnt = 4;            // 글자개수 : 4개
char str[5] = {};                         // 글자 조합 경우
int cnt = 0;                              // 경우의 수

void recursive(int idx)
{
        if (idx == length)
        {
                //cout << str << endl;
                cnt++;
                return;
        }

        for (int i = 0; i < charCnt; ++i)
        {
                bool isPossible = true;
                if (idx != 0)
                {
                        if (str[idx - 1] == 'T')
                        {
                                if (inputs[i] == 'B')
                                        isPossible = false;
                                else
                                        isPossible = true;
                        }
                        else if (str[idx - 1] == 'B')
                        {
                                if (inputs[i] == 'T')
                                        isPossible = false;
                                else
                                        isPossible = true;
                        }
                        else
                                isPossible = true;
                }

                if (isPossible)
                {
                        str[idx] = inputs[i];
```

```
                        recursive(idx + 1);
                        str[idx] = '\0';

                }
        }
}

int main(void)
{
        cin >> inputs;

        int initIdx = 0;
        recursive(initIdx);
        cout << "total case : " << cnt;
        return 0;
}
```

Level23 ABC초콜릿 [난이도 : 4]

문제 3번 [숙제 목록보기]

A /B /C 세 종류의 초콜릿이 있습니다

이 중 n개의 초콜릿을 선택하려고 하는데

3개 연속으로 같은 알파벳의 초콜릿을 선택하면 안됩니다.

가져갈 n개의 초콜릿 개수를 입력받고, 나올수 있는 총 가짓수를 출력해주세요.

(재귀호출을 이용해서 풀어주세요)

ex) 3개의 초콜릿을 선택한다고 한다면

AAA --> 불가능

AAB --> 가능

AAC --> 가능

ABA --> 가능

...

CCC -> 불가능

총 24가지

입력 예제

```
3
```

출력 결과

```
24
```

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

char chocoes[4] = "ABC";
const int chocoSort = 3; // 초코의 종류
int maxCnt = 0;                          // 최대 초코 개수
int caseCnt = 0;                 // 총 경우의 수
char bucket[10] = {};

void recursive(int cnt)
{
        if (cnt == maxCnt)
        {
                cout << bucket << endl;
                ++caseCnt;
                return;
        }

        bool isPossible = true;
        for (int i = 0; i < chocoSort; ++i)
        {
                // 3개 연속 같은 초코를 고르는지 검사
                if (cnt > 1)             // 3개째를 고르는 순간부터 확인
                {
                        if (bucket[cnt - 1] == bucket[cnt - 2] && bucket[cnt - 1] == chocoes[i])
                                isPossible = false;
                        else
                                isPossible = true;
                }

                if (isPossible)
                {
                        bucket[cnt] = chocoes[i];
                        recursive(cnt + 1);
                        bucket[cnt] = '\0';
                }
        }

}
```

```
int main(void)
{
	cin >> maxCnt;

	int cnt = 0;
	recursive(cnt);
	cout << "total case : " << caseCnt;

	return 0;
}
```

Level23 산타소년단 [난이도 : 5]

문제 4번 [숙제 목록보기]

인기그룹 산타소년단이 있습니다.

맴버 다섯명의 이름은 각각 B T S K R 입니다.

이 그룹에서 n명을 순서대로 뽑아서 새로운 소규모 그룹을 만드려고 합니다.

첫 번째 뽑인사람이 리더이며,

두 번째 이후부터는 세컨드, 서드 등등 역할이 주어집니다.

맴버 중 방송국 국장 아들인 S군은 새로운 소규모 팀에 들어있어야 합니다.

n을 입력받으세요.

그리고 n명을 뽑아 맴버를 구성하려고 할 때,

나올 수 있는 순열의 수를 Counting해서 출력 해 주세요.

ex) 3

BTS

BST

BSK

BSR

BKS

BRS

TBS

TSB

TSK

TSR

TKS

TRS

SBT

SBK

SBR

STB

STK

STR

SKB

SKT

SKR

SRB

SRT

SRK

KBS

KTS

KSB

KST

KSR

KRS

RBS

RTS

RSB

RST

RSK

RKS

총 36개

[힌트]

cnt++ 하기 전, via['S'의 index] == 1 인지 확인하면

S가 포함되었는지 알 수 있음

입력 예제

```
3
```

출력 결과

```
36
```

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

char members[6] = "BTSKR";
char newTeam[6] = {};
const int people = 5;                    // 뽑을 수 있는 사람 수 : 5명
int newTeamMaxMember = 0;        // 새로운 팀의 최대 인원 수 : n명
int caseCnt = 0;                         // 총 경우의 수
int isPicked[5] = {};                    // 해당 멤버가 이미 뽑혔는지 확인

bool checkS()
{
        for (int k = 0; k < newTeamMaxMember; ++k)
        {
                if (newTeam[k] == 'S')
                        return true;
        }
        return false;
}

void recursive(int idx)
{
        if (idx == newTeamMaxMember)
        {
                if (checkS())    // 함수는 뒤에 '()'을 붙여줘야한다.
                {
                        //cout << newTeam << endl;
                        ++caseCnt;
                }
                return;
        }

        for (int i = 0; i < people; ++i)
        {
                if (isPicked[i] == 0)
                {
                        newTeam[idx] = members[i];
                        isPicked[i] = 1;
                        recursive(idx + 1);
                        newTeam[idx] = '\0';
                        isPicked[i] = 0;
                }
        }
}
```

```
int main(void)
{
	cin >> newTeamMaxMember;

	int initIdx = 0;
	recursive(initIdx);
	cout << "total case : " << caseCnt;

	return 0;
}
```

Level23 미안하다 친구야 [난이도 : 4]

문제 5번 [숙제 목록보기]

'E', 'W', 'A', 'B', 'C' 라는 친구들이 놀이기구를 타려고 합니다.

이 놀이기구는 가장 앞좌석부터 뒷좌석까지 4명이 탈 수 있는 보트입니다.

다섯명의 친구들 중 탈 순서를 정해야 하는데,

한명을 제외시켜야 합니다.

제외시킬 친구를 입력받고 (문자 1개입력)

이 친구를 제외한 모든 순열을 출력 해 주세요.

입력 예제

```
E
```

출력 결과

```
WABC
WACB
WBAC
WBCA
WCAB
WCBA
AWBC
AWCB
ABWC
ABCW
ACWB
ACBW
BWAC
BWCA
BAWC
BACW
```

BCWA

BCAW

CWAB

CWBA

CAWB

CABW

CBWA

CBAW

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

char friends[5] = { 'E', 'W', 'A', 'B', 'C' };     // 친구 목록
const int people = 5;                              // 총 사람의 수
const int seat = 4;                                // 앉을 수 있는 자리 수
char boat[seat + 1] = {};                          // 보트
int isSeat[5] = {};                                // 승차 여부
char excep = '\0';                                 // 승차 제외 인물
int caseCnt = 0;                                   // 경우의 수

/*
        이유는 알 수 없으나
        excep 의 값을 'E'로 입력받으면
        할당하지 않은 boat[4] 의 값이 E가 되며
        boat 배열을 출력하면 모든 경우 뒤에 E가 따라 나온다.
        때문에 boat[seat + 1] 로 할당하여 boat[4]의 값을 null로 초기화
*/
void recursive(int seatNum)
{
        if (seatNum == seat)
        {
                cout << boat << endl;
                ++caseCnt;
                return;
        }

        for (int i = 0; i < people; ++i)
        {
                if (isSeat[i] == 0 && friends[i] != excep)
                {
                        boat[seatNum] = friends[i];
                        isSeat[i] = 1;
                        recursive(seatNum + 1);
                        boat[seatNum] = '\0';
                        isSeat[i] = 0;
                }
        }
}
```

```
int main(void)
{
	cin >> excep;

	int initSeat = 0;
	recursive(initSeat);
	cout << "total case : " << caseCnt;

	return 0;
}
```

Level23 다섯종류의 숫자카드 [난이도 : 5]

문제 6번 [숙제 목록보기]

다섯 종류의 카드를 입력받습니다. ('0' ~ '9')

각각의 카드들은 다량으로 쌓여있습니다.

다섯 종류의 숫자 카드에서 **4장을** 뽑으려고 합니다.

뽑을 때마다 전에 뽑았던 카드번호와 간격이 **3이하로 차이나는**

중복순열이 몇 가지인지 출력하세요.

재귀호출을 이용해서 풀어주세요

ex)

카드종류가 1/2/3/4/5 일때

1111 : OK

1112 : OK

1113 : OK

1114 : OK

1115 : [NO]

1121 : OK

...

no count 숫자 : 1251, 5123 .. 등등

총 497가지

[힌트]

path[level - 1] 와 path[level] 의 절대값이 3 차이가 나는지 확인

입력 예제

```
12345
```

출력 결과

```
497
```

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

const int cardCnt = 5;                    // 총 카드의 개수
const int cardPickCnt = 4;                // 뽑을 수 있는 카드의 개수
char cardCase[cardCnt] = {};              // 카드 종류
const int interval = 3;                   // 기준 간격
int caseCnt = 0;                          // 조건에 맞는 경우의 수
char cardPickCase[cardPickCnt + 1] = {};  // 카드 뽑은 현황

void recursive(int idx)
{
	if (idx == cardPickCnt)
	{
		cout << cardPickCase << endl;
		++caseCnt;
		return;
	}

	for (int i = 0; i < cardCnt; ++i)
	{
		// 앞에 뽑은 카드와 지금 뽑을 카드의 차이가 3이하가 아닌 경우 해당 경우는 생략
		if (idx != 0)
		{
			int sub = cardPickCase[idx - 1] - cardCase[i];
			if (sub < 0)
				sub *= (-1);
			if (sub > interval)
				continue;
		}
		cardPickCase[idx] = cardCase[i];
		recursive(idx + 1);
		cardPickCase[idx] = '\0';
	}
}

int main(void)
{
	cin >> cardCase;

	int initIdx = 0;
	recursive(initIdx);
	cout << "total case : " << caseCnt;

	return 0;
}
```