

1. 아래와 같이 학생(Student) 클래스를 입력하고, 상속 받을 학부생 클래스(Undergraduate)와 조교 클래스(Graduate)를 만든 후에 정보를 출력해주세요.

- 1) 학생의 기본정보는 이름, 학번, 과, 학년, 이수점수가 있습니다.
- 2) 학부생의 정보는 추가적으로 소속 동아리명이 있습니다.
- 3) 조교의 경우에는 조교유형과 장학비율이 있으며, 조교 유형의 경우 연구조교와 교육조교로 나뉩니다.
- 4) Base Class인 Student는 print 함수 사용 시 "I'm student"를 출력합니다.

```
int main(void)
{
    Student Std;
    Graduate Kang("강석원", 2018311025, "컴퓨터공학과", 1, 18, false, 0.5);
    UnderGraduate Hong("홍길동", 20141111, "컴퓨터공학과", 2, 80, "알고리즘동아리");

    Std.print();
    Kang.print();
    Hong.print();

    return 0;
}
```

[참조]

```
class Student {
```

```
private:
```

```
    string name;           //이름
    int id;                 //학번
    string department;      //과
    int grade;              //학년
    int credit;             //이수학점
```

```
public:
```

```
    Student(string n = "default", unsigned int i = 0, string d = "default", int g = 0,
int c = 0) : name(n), id(i), department(d), grade(g), credit(c) {}
```

```
    /* Getter, Setter 함수를 구현해주세요 */
    /* 또한 virtual을 활용해서 print함수를 만들어주세요 */
```

```
};
```

1 - 출력화면

```
C:\> 선택 C:\WINDOWS\system32\cmd.exe

I'm student

=====
이름      : 강석원
학번      : 2018311025
소속학과  : 컴퓨터공학과
학년      : 1
이수학점  : 18
조교유형  : 연구조교
장학비율  : 0.5
=====

이름      : 홍길동
학번      : 20141111
소속학과  : 컴퓨터공학과
학년      : 2
이수학점  : 80
동아리명  : 알고리즘동아리
=====
계속하려면 아무 키나 누르십시오 . . .
```

2. 애너그램을 만들기위해 삭제해야할 최소 문자수를 출력하세요.

두 영단어의 순서를 바꾸어서 같아질 수 있다면, 두 단어는 애너그램 관계입니다. 예를들면 apple은 eplpa는 서로 애너그램 관계에 있습니다. 반대로 apple과 elpaa는 애너그램 관계가 될 수 없습니다.

그렇다면, 두 개의 영단어가 주어졌을 때, 두 단어가 애너그램이 되도록 삭제해야할 문자수를 카운트해서 출력해주세요.

2 - 출력 예시

```
cmd 실행 C:\WINDOWS\system32\cmd.exe
두 단어를 입력해주세요 : XXYYBB AABBC
8
```

- 여기서는 X,X,Y,Y,A,A,C,C 총 8개의 문자가 삭제되어야 두 단어가 애너그램(BB)이 될 수 있습니다.

```
cmd 실행 C:\WINDOWS\system32\cmd.exe
두 단어를 입력해주세요 : AABBCDD ABCDE
5
```

3. 숫자 야구를 구현해주세요.

- 1) 컴퓨터는 정답으로 쓰일 1 ~ 9 범위 내 세 개의 숫자를 임의로 선택합니다.
- 2) 유저는 매 턴 숫자 세 개를 입력합니다.
- 3) 컴퓨터는 유저가 입력한 값에서 숫자는 맞지만 위치가 틀릴 경우에는 Ball을, 위치도 맞고 숫자도 맞을 경우는 Strike를 카운트하여 출력합니다.
- 4) Strike와 Ball이 각각 0일 경우에는 Out을 출력합니다.
- 4) 9번의 시도에도 정답을 맞히지 못하는 경우 패배합니다.
- 5) 정답과 입력은 중복된 값을 사용할 수 없으며 1 ~ 9 범위 내에서 정해야합니다.

```
int main() {  
    int input[3];  
    int answer[3];  
  
    /* 1 ~ 9 범위 내 임의의 숫자 세개를 선정*/  
  
    startGame(input, answer);  
  
    return 0;  
}
```

3 - 출력예시

```
CA. 실행 C:\WINDOWS\system32\cmd.exe  
===== 1 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 2 3  
0 Strike, 1 Ball  
  
===== 2 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 4 5 6  
Out!!  
  
===== 3 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 8 9  
0 Strike, 2 Ball
```

```
===== 4 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 2 9  
0 Strike, 2 Ball  
  
===== 5 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 7 3 9  
1 Strike, 2 Ball  
  
===== 6 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 9 3 7  
3 Strike, 0 Ball  
  
정답입니다!  
계속하려면 아무 키나 누르십시오 . . .
```

```
===== 9 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 2 3  
0 Strike, 1 Ball  
  
패배했습니다.  
정답은 4 3 7 입니다.  
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\WINDOWS\system32\cmd.exe  
===== 1 =====  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 1 1 1  
중복된 숫자를 입력하셨습니다.  
1 ~ 9 범위의 숫자 세개를 입력해주세요 : 10 10 10  
1 ~ 9 범위 내의 숫자를 입력하여 주세요.
```

4. Stack Class를 구현하세요.

변수(private):

```
int* p_list; // 정수형 변수들을 가지는 배열
int size; //현재 저장된 변수들의 개수
int MAX_SIZE; // 최대로 저장할 수 있는 p_list의 크기
```

함수(public);

Stack(int _MAX_SIZE = 1000) //생성자: p_list의 크기를 MAX_SIZE만큼 동적 할당.

~Stack() // 소멸자: p_list의 동적 할당을 해제

int find_index(int _item) // p_list에서 _item과 동일한 값이 있는지 검색 후 발견시 index를 반환한다 만약 발견하지 못하면 -1을 반환한다

void push(int_item) // 입력item을 p_list의 끝에 저장한다. 만약 _item과 동일한 값이 p_list에 존재할 경우 p_list에 _입력 item을 추가하지 않는다. (힌트: find_index 함수를 사용해서 중복된 값이 p_list에 있는지 조사후 없는 경우에 입력 item을 p_list에 추가). size가 MAX_SIZE보다 크면 item을 추가하지 않는다.("Error: out of memory"출력)

int pop() // p_list에 있는 마지막 item을 제거한다음 그 아이템을 return한다 (힌트:size 값을 줄이면 p_list의 아이템을 제거한 것과 동일한 효과) size가 0일 때는 item을 제거하지 않는다. ("Error: No item exists in the list"출력)

void print() const // Stack 객체의 item들을 출력한다

int get_size() //Stack 객체의 크기를 출력한다

int get_item(int _index) // p_list의 해당 index에 있는 item 값을 리턴한다.

[아래 기능을 추가하세요]

- (s1 += s2) s1에 s1과 s2의 모든 item을 합친다. 이때 s1 끝에 s2의 아이템을 추가하며, 중복된 값을 가지는 item은 추가되지 않는다.

- (s1==s2) s1과 s2의 size와 item이 모두 동일한 경우 true를, 그렇지 않은경우 false를 출력한다. 이때, item이 저장된 순서도 같아야 한다.

시작 코드

#do not modify below

```
int main()
```

```
{
```

```
    Stack s1, s2;
```

```
    s1.push(1);
```

```
    s1.push(2);
```

```
    s1.push(3);
```

```

s2.push(1);
s2.push(2);
s2.push(5);

s1.print();
s2.print();

s1 += s2;

s1.print();
s2.print();
cout << "s1 == s2 ? " << (s1 == s2) << endl;

s1.pop(); // 5 out
s1.pop(); // 3 out
s2.pop(); // 5 out
cout << "s1 == s2 ? " << (s1 == s2) << endl;
s2.pop();
s2.pop();
s2.pop();

return 0;
}

```

```

Items in the list : 1, 2, 3,
Items in the list : 1, 2, 5,
Items in the list : 1, 2, 3, 5,
Items in the list : 1, 2, 5,
s1 == s2 ? 0
s1 == s2 ? 1
Error: No item exist in the list

Process finished with exit code 0

```