

1. 세명의 User 정보(ID, Password)를 입력받은 후에 반복하여 로그인하는 프로그램을 구현 하세요.

- 1) 입력받은 User는 세명입니다.
- 2) 로그인시에 ID에 “종료”를 입력하면 프로그램이 종료됩니다.
- 3) ID는 중복 될 수 없습니다. 중복된 ID를 입력할 경우에 종료됩니다.

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

int main() {
    User user[3];
    string id, password, searchId, searchPassword;

    for (int i = 0; i < 3; i++) {
        /* User 정보를 입력받음 */
    }

    while (1) {
        /* LogIn 기능을 구현, ID에 “종료” 입력시 프로그램 종료 */
    }
}
```

## 1 - 출력화면

CAJ 실행 C:\WINDOWS\system32\cmd.exe

```
===== 1 =====
id : 홍길동
password : 1234
=====

===== 2 =====
id : 민경훈
password : 5678
=====

===== 3 =====
id : 김민수
password : 1357
=====

===== LogIn =====

Id : 홍길동
Password : 1234
로그인 되었습니다.
=====

===== LogIn =====

Id : 민경훈
Password : 2345
잘못된 ID거나 PASSWORD 입니다.
=====

===== LogIn =====

Id : 종료
종료하겠습니다.
계속하려면 아무 키나 누르십시오 . . .
```

CAJ 실행 C:\WINDOWS\system32\cmd.exe

```
===== 1 =====
id : 민경훈
password : 1234
=====

===== 2 =====
id : 민경훈
이미 존재하는 ID입니다.
종료합니다.
계속하려면 아무 키나 누르십시오 . . .
```

2. 좌표값 두 개를 입력받고 두 좌표 사이의 거리를 출력하세요.

- 1) 좌표값은 초기화시에 값을 따로 주지 않으면  $x = 0$ ,  $y = 0$ 으로 초기화합니다.
- 2) “좌표 - 좌표” 연산자를 Operator overloading을 활용하여 선언하세요.  
·  $(x1, y1) - (x2, y2) = (x1 - x2, y1 - y2)$ 를 클래스 내에서 구현합니다.
- 3) “좌표 \* 좌표” 연산자를 Operator overloading을 활용하여 선언하세요.  
·  $(x1, y1) * (x2, y2) = (x1 * x2, y1 * y2)$ 를 클래스 내에서 구현합니다.
- 4) 2번과 3번을 활용하여 두 좌표 사이의 거리를 구하세요.
- 5) 제곱근 사용을 위해 `#include <cmath>`를 선언해야합니다.

```
int main() {
    int x1=0, y1=0, x2=0, y2=0;
    Point *pP1, *pP2, *pP3;

    cout << "첫번째 좌표[x1, y1]를 입력하세요 : ";
    cin >> x1 >> y1;

    cout << "두번째 좌표[x1, y1]를 입력하세요 : ";
    cin >> x2 >> y2;

    pP1 = new Point(x1, y1);
    pP2 = new Point(x2, y2);

    pP3 = new Point(); //x,y가 0으로 초기화

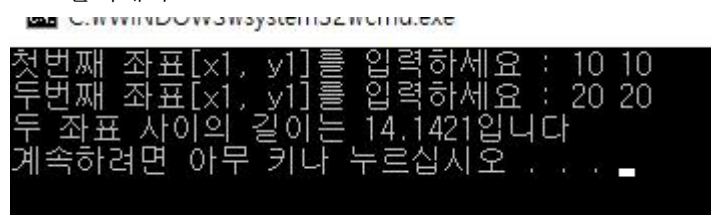
    /* 아래의 방식으로도 x, y값 설정이 가능해야합니다. */
    //pP1->setPoint(x1, y1);
    //pP1->setPoint(x2, y2);
    /*******/

    *pP3 = (*pP1 - *pP2) * (*pP1 - *pP2);

    /* pP3을 활용하여 거리값을 구하세요 */
    cout << "두 좌표 사이의 길이는 " << /* 결과 값 */ << "입니다" << endl;;

    return 0;
}
```

2 - 출력예시



```
C:\Windows\system32\cmd.exe
첫번째 좌표[x1, y1]를 입력하세요 : 10 10
두번째 좌표[x1, y1]를 입력하세요 : 20 20
두 좌표 사이의 길이는 14.1421입니다
계속하려면 아무 키나 누르십시오 . . .
```

3. 학생들끼리 송금하는 프로그램을 작성하세요.

1. 한번 송금을 할 때, 돈을 보내는 학생의 전 재산(balance)를 송금합니다.
2. Account(계좌) class는 멤버변수로  
string name(이름), string id(학번), int balance 를 갖고있도록 합니다.
3. 사용자로부터 돈을 보낼 학생의 학번과 돈을 받을 학생의 학번을 입력받습니다. 이 때,  
다음의 경우에 대해서는 다시 입력 받도록 합니다.
  - 1) 돈을 보낼 학생과 돈을 받을 학생의 학번이 동일한 경우
  - 2) 보낼 학생 혹은 받을 학생의 학번이 존재하지 않는 경우
  - 3) 보낼 학생의 잔액이 0인 경우
4. setBalance 함수를 사용하지 않고, Operator overloading(+, -)를 이용하여 송금후 보낸 학생의 잔액과 받은 학생의 잔액을 계산하세요.
5. 송금이 완료 된후, Operator overloading(<<)를 이용하여 돈을 보낸 학생과 받은 학생의 계좌를 출력하세요.
6. 사용자로부터 돈을 보낼 학생의 학번을 “종료”라고 입력받았을 경우 Operator overloading(<<)를 이용하여 모든 학생의 계좌를 출력하세요.

```
int main() {  
  
    Account acnt[3] = {  
        Account("신은섭", "2019204212", 10000),  
        Account("김유민", "2012311060", 0),  
        Account("이강호", "2011102214", 5000),  
    };  
  
    /* 구현 */  
  
}
```

### 3 - 출력예시

[illegible]

4. 학생 계좌 정보를 입력받고 학생 계좌 정보들을 모두 삭제하여 삭제된 계좌들의 잔액 총합을 출력하는 프로그램을 작성하세요.

1. 총 학생 수를 사용자로부터 입력받습니다.
2. 학생 계좌 정보 Account class는 학번, 이름, 잔액을 멤버 변수로 가지고 있습니다.
3. Account class 배열을 동적으로 생성합니다.
4. 각 학생의 학번, 이름, 잔액을 입력받습니다. 이 때, 중복된 학번을 입력받으면 프로그램을 종료합니다.
5. 모두 입력 받은 후, Account class 배열을 delete하여 모든 학생 계좌의 잔액 총합을 출력하세요.

Hint: Account class 내 멤버 변수에서 static 변수를 사용, Account class의 소멸자를 적절히 이용

#### 4- 출력 예시

```
총 학생 수 입력: 3
1번째 학생 계좌 입력 : 학번 : 2015
이름 : 정준현
잔액 : 2000
=====
2번째 학생 계좌 입력 : 학번 : 2019
이름 : 노혜린
잔액 : 2500
=====
3번째 학생 계좌 입력 : 학번 : 2016
이름 : 홍길동
잔액 : 3000
=====
회수된 금액 : 7500
```