

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="";
    string searchId = "", searchPassword = "";

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

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

## 1 - 출력화면

CA> 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 . . .
```

CA> 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);

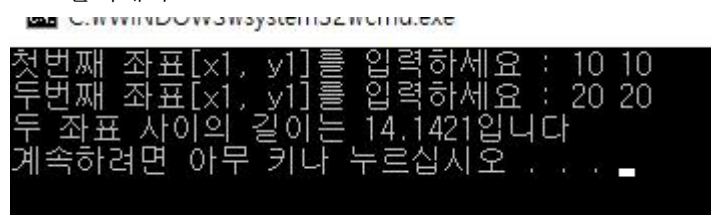
    pP3 = new Point(); //x,y가 0으로 초기화

    /* 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;;
    delete pP1, pP2, pP3;
    return 0;
}
```

2 - 출력예시



3. LAB#06에서 피보나치 수열을 출력하는 함수를 만들었습니다. 이번엔 피보나치 수열을 활용하여 시저 암호문을 만들어보세요.

1) 시저 암호문(Caesar cipher)는 간단한 암호문의 일종으로, 일정한 숫자만큼 알파벳을 밀어서 문장을 치환하는 암호화 알고리즘입니다.

예시) abcde, 1 = bcdef / xyz, 2 = zab

2) 피보나치 수열에 맞춰서 문장의 문자를 밀어야합니다.

예시) 입력한 문장(abcde) + 피보나치 수열(0 1 1 2 3) -> acdfh

3) 입력은 알파벳만 받으며, 만약 알파벳 외에 다른 문자를 받을 경우 프로그램을 종료합니다.

4) 문장 내에 대문자를 모두 소문자로 치환합니다. 치환 함수는 내장된 함수가 아니라 직접 구현하세요.

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

int fibonacci(int);
void lowercase(string& plainText);

int main()
{
    int n;
    int *p;

    while (1) {
        string plainText = "";
        string cryptoText = "";
        cout << "input : ";
        cin >> plainText;

        /* 알파벳이 아닌 문장 필터링 */

        lowercase(plainText);
        cout << "입력한 평문 : " << plainText << endl;

        /* LAB#06의 피보나치 수열을 활용하세요. */

        cout << cryptoText << endl;
        cout << "\n";
    }
    return 0;
}
```

### 3 - 출력예시

실행 : C:\WINDOWS\system32\cmd.exe

```
input : HELLO
입력한 평문 : hello
암호화된 문장 : hfmnr

input : HELLoWorld
입력한 평문 : helloworld
암호화된 문장 : hfmnrbwegl

input : abcde
입력한 평문 : abcde
암호화된 문장 : acdfh

input : abcdefghijklmnopqrs
입력한 평문 : abcdefghijklmnopqrs
암호화된 문장 : acdfhkoudrnwamfjbaq

input : 안녕하세요
알파벳이 아닌 문자가 입력되었습니다.
프로그램을 종료합니다.
계속하려면 아무 키나 누르십시오 . . .
```

#### 4. [사진관리 프로그램 작성을 위한 단계 2]

lab#6에서 작성한 프로그램을 다음과 같이 확장한다. 메인함수는 [참고 1]과 동일하게 한다

- ReadRecordFromKB() 함수에서 파일명을 자동으로 생성하도록 수정한다.(아래 [참조 2]을 참조)
- 사진파일명(photoName)과 이벤트 명(eventName)을 저장하기 위한 vector 객체들을 선언하고 여기에 읽어들이 파일정보를 vector의 멤버함수를 이용하여 저장한다.
- Vector 객체들에 저장된 정보를 파일로 출력한다.
- 파일에 저장된 정보를 읽어서 vector에 저장한다.
- 파일명으로 사진파일을 검색하고 관련 정보를 화면에 출력한다.
- 파일명으로 사진파일을 검색하고 관련 정보를 vector list에서 삭제한다.

보다 구체적인 내용은 아래와 같다.

1)키보드로부터 하나의 사진파일 레코드를 입력받아서 vector list에 저장하는 함수를 작성한다.

```
int AddRecordToList(vector<string> &pName, vector<string> &eventName)
```

1. vector에 저장된 모든 정보를 파일로 출력하는 함수를 작성한다.

```
int WriteAllToFile(vector<string> &nameList, vector<string> &eventList)
```

2. 파일에 저장된 모든 정보를 읽어서 vector에 저장하는 함수를 작성한다.

```
int ReadAllFromFile(vector<string> &nameList, vector<string> &eventList)
```

3. 사진파일명을 입력받아서 vector list에서 해당 레코드를 찾아서 index를 리턴하는 함수를 작성한다.

```
int SearchByPrimKey (vector<string> v, string inName)
```

4. Vector list에 저장된 모든 정보를 화면에 출력한다.

```
void DisplayAllOnScreen(vector<string> &nameList, vector<string> &eventList)
```

5. 키보드로부터 파일명을 입력받아서 SearchByPrimKey()를 이용하여 해당파일 인덱스를 찾고 이것을 이용하여 해당 파일(또는 레코드)정보를 모두 화면에 출력하는 함수를 작성한다.

```
void FindAndDisplay(vector<string> &nameList, vector<string> &eventList)
```

6. 키보드로부터 파일명을 입력받아서 SearchByPrimKey()를 이용하여 파일 인덱스를 찾아서 해당 레코드(photoName과 eventName)을 vector list로부터 삭제하는 함수를 작성한다. (vector 멤버함수 사용)

```
void DeleteByName(vector<string> &nameList, vector<string> &eventList)
```

7. [참조 2]를 바탕으로 다음과 같이 main 함수를 작성한다.

- 파일명을 저장하기 위한 vector 객체 생성
- 이벤트명을 저장하기 위한 vector 객체 생성
- 명령을 선택하여 수행(참조 3를 바탕으로 작성)

#### [참조 1] 프로그램 main 함수

```
int main()
{
    vector<string> nameList(20);
    vector<string> eventList(20);
    nameList.clear();
    eventList.clear();
    Run(nameList, eventList);

    return 1;
}
```

#### [참조 2] 현재 시간을 문자열로 변환해주는 코드

/\* C++ program to read current time

```
==== The structure type holding the date and time =====
struct tm {
    int tm_sec;    // seconds of minutes from 0 to 61
    int tm_min;    // minutes of hour from 0 to 59
    int tm_hour;   // hours of day from 0 to 24
    int tm_mday;   // day of month from 1 to 31
    int tm_mon;    // month of year from 0 to 11
    int tm_year;   // year since 1900
    int tm_wday;   // days since sunday
    int tm_yday;   // days since January 1st
    int tm_isdst;  // hours of daylight savings time
}
*/
```

```

#include<iostream>
#include<ctime>
#include <string>
#include <sstream>
#include <iomanip>
#include <stdio.h>
using namespace std;
int main()
{
    // Declaring argument for time()
    time_t tt;

    // Declaring variable to store return value of localtime()
    struct tm * ti;

    // Applying time()
    time (&tt);

    // Using localtime()
    ti = localtime(&tt);

    cout << "Current Day, Date and Time is = " << asctime(ti)<<endl;
    cout<< "year: " <<ti->tm_year+1900<< " month : " <<ti->tm_mon+1<< " day : " <<ti->tm_mday
    << " hours : " << ti->tm_hour << " minutes: " << ti->tm_min<< " second: " <<ti->tm_sec<<endl;
    cout<< "year: " <<ti->tm_year+1900<< " month : " <<ti->tm_mon+1<< " day : " <<ti->tm_mday
    << " hours : " << ti->tm_hour << " minutes: " << ti->tm_min<< " second: " <<ti->tm_sec<<endl;

    // generate containing the 14 digit number in the following format: yyyymmddhhmmss
    stringstream ss;
    ss << setw(4)<<ti->tm_year+1900<<setfill('0')<<setw(2)<<ti->tm_mon+1<<setfill('0')
    <<setw(2)<<ti->tm_mday<<setfill('0')<<setw(2)<<ti->tm_hour<<setfill('0')<<setw(2)
    <<ti->tm_min<<setfill('0')<<setw(2)<<ti->tm_sec<<"W0";

    string ymd=ss.str();
    cout<<"Size of the string: "<<ymd.size()<<"    string: " <<ymd<<endl;

    int temp;
    cin>>temp;
    return 0;
}

```

### [참조 3]

```

<< main 함수 부분>>
int command;
while((command=GetCommand())){
    switch (command) {
        case 0 :
            return;
        case 1: // Read in a record and add to the list
            AddRecordToList(nameList, eventList);
            break;
        case 2 : // Read all the records from file
            ReadAllFromFile(nameList, eventList);
            break;
        case 3: // Save all photo records in file.
            WriteAllToFile(nameList, eventList);
            break;
        case 4: // Display all the records on the screen
            DisplayAllOnScreen(nameList, eventList);
            break;
        case 5: // Get a file name from keyboard, find it, and display on screen
            FindAndDisplay(nameList, eventList);
            break;
        case 6: // Get a file name from keyboard, find it, and delet it from list
            DeleteByName(nameList, eventList);
            break;
        default:
            cout<< "Invalid operation Wn";
            break;
    }
}

```

<<GetCommand() 함수>>  
int GetCommand()

```

{
    int temp;
    while (1)
    {
        cout << endl << endl;
        cout<<"\t--- ID - Command ----"<<endl;
        cout<<"\t 1. Add a new record to list"<<endl;
        cout<<"\t 2. Read all records from disk"<<endl;
        cout<<"\t 3. Write all records to disk"<<endl;
        cout<<"\t 4. Display all record on screen"<<endl;
        cout<<"\t 5. Search a record by file name and display on screen"<<endl;
        cout<<"\t 6. Search a record by file name and delet it from list"<<endl;
        cout<<"\t 0. Quit"<<endl<<endl;
        cout << "\t Choose a Command--> ";
        if (cin.fail() || !(cin >> temp) || temp<0 || temp>6) // input fail or input value is out
        of bound
        {
            cout<<"\t   ### Invalid Command Number. Select again   ###"<<endl;

            cin.clear(); //clear flag
            cin.ignore(INT_MAX, '\n'); //empty input buffer
        }
        else {
            cin.clear(); // clear flag
            cin.ignore(INT_MAX, '\n'); // empty input buffer
            return temp;
        }
    }
}

```



[참조 3] 실행화면

```
***** [ Records in the List ] *****
Record 0 : sss , ffff
Record 1 : ggg , dmfm,d
Record 2 : kk , dfkjkd

--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit

Choose a Command--> 1
Read a photo file(phot name and event name)
Photo Name --> sdkfkd
Photo Name --> 2017졸업식

--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit

Choose a Command--> 5

***** [ Search And Display ] *****
Enter a file name to search --> ggg
Record : ggg , dmfm,d

--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit

Choose a Command--> 3

--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit

Choose a Command--> 2
```

Choose a Command--> 2

```
--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit
```

Choose a Command--> 4

```
***** [ Records in the List ] *****
Record 0 : sss , ffff
Record 1 : ggg , dmfm,d
Record 2 : kk , dfkjkd
Record 3 : sdkfkd , 2017졸업식
```

```
--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit
```

Choose a Command--> 6

```
***** [ Delete a record by name ] *****
Enter a file name to search --> ggg
```

```
--- ID - Command -----
1. Add a new record to list
2. Read photo list from disk
3. Write photo list to disk
4. Display list on screen
5. Retrieve by file name
6. Delete by file name
0. Quit
```

Choose a Command--> 4

```
***** [ Records in the List ] *****
Record 0 : sss , ffff
Record 1 : kk , dfkjkd
Record 2 : sdkfkd , 2017졸업식
```