

Exam #2



경희대학교
KYUNG HEE UNIVERSITY

- 주어진 Exam02_01.cpp 파일을 보고 2~50 범위 내의 정수의 소수여부를 판정하여 출력하는 프로그램을 작성하여라.

목표: 2~50 범위 내의 모든 정수의 소수 여부를 판정하여 출력한다.

- 소수 여부를 판단할 수 있는 bool 형 변수를 이용
- 아래 소수의 정의 및 판정 방법을 확인하여 코드를 작성.
- 아래 결과와 같이 출력되도록 소스를 작성함

소수의 정의 및 판정 방법

- 정의** : 1과 자기 자신만으로 나누어 떨어지는 1보다 큰 양의 정수
- 소수의 예** : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 등은 소수이다.
- 소수 판정 방법** : 자연수 n 이 소수인지 아닌지를 판정하려면 $2 \leq p \leq \sqrt{n}$ 인 범위에 있는 모든 소수 p 로 n 을 나누어 보아, 나누어 떨어지지 않으면 소수이고, 나누어 떨어지면 합성수이다.
 (힌트 : 나눗셈의 나머지가 0인지 조사하여 나누어 떨어지는가를 확인)

output

```

2는 소수입니다.
3는 소수입니다.
4는 소수가 아닙니다.
5는 소수입니다.
6는 소수가 아닙니다.
7는 소수입니다.
8는 소수가 아닙니다.
9는 소수가 아닙니다.
10는 소수가 아닙니다.
11는 소수입니다.
12는 소수가 아닙니다.
13는 소수입니다.
14는 소수가 아닙니다.
15는 소수가 아닙니다.
16는 소수가 아닙니다.
17는 소수입니다.

```

● Exam02_01.cpp 파일

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    bool isPrime = true; // 소수 판정을 저장하기 위한 bool형 변수
    for(int i=2; i<50; i++)
    {
        // 정수 2부터 50에 대해 소수 여부를 판정하기 위한 코드 작성
    }
    return 0;
}
```

1) 제곱근 함수(sqrt()) 설명

- 함수 원형 : Double sqrt(double x);
- 헤더 파일 : "math.h"
- 함수 의미 : 매개변수에 전달되는 x값의 제곱근을 반환한다.
- 예제 : sqrt(9)는 3을 반환한다.

2) 나머지 연산자(Modulus Operator) % 설명

- op1 % op2 는 op1을 op2로 나누었을 때의 나머지를 의미함
- 예제 : 7 % 3 의 결과는 1

- 주어진 Exam02_02.cpp 파일을 보고 2차원 배열(행렬)의 값을 입력 받아 행렬 요소의 최대값과 그 위치(행과 열)를 찾아 출력하는 프로그램을 작성하여라.

목표: 행렬의 값을 사용자로부터 입력 받아 행렬 요소의 최대값과 그 위치를 출력하여라.

- 사용자로부터 2차원 배열의 행과 열의 개수를 입력받는다.
- 각행의 배열 요소값을 사용자로부터 입력받아 저장.
- 아래 정의된 MaxArray함수를 이용하여 최대값과 그 위치를 얻어온다.
- 아래 결과와 같이 출력되도록 소스를 작성함

MaxArray 함수

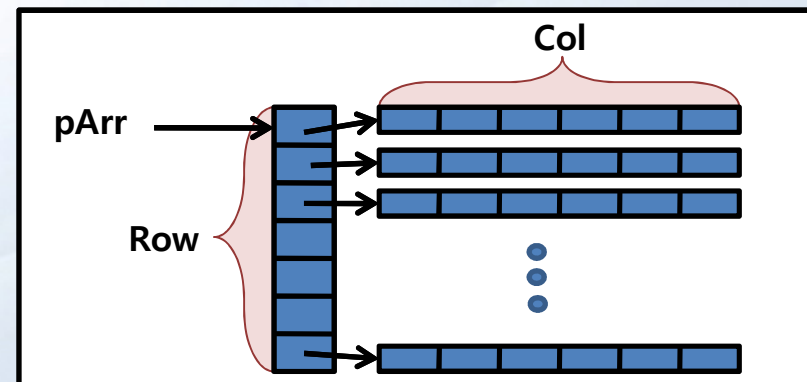
- 함수 원형 : void MaxArray(int **pArr, int row, int col, int *result);
- 각 매개변수의 의미 :
 - pArr : 원본 행렬
 - Row : 행렬의 행의 수
 - Col : 행렬의 열의 수
 - Result : 결과값(최대값과 위치)이 저장될 행렬.
 - Result[0] = 행렬의 최대값 저장
 - Result[1] = 최대값이 위치하는 행의 번호
 - Result[2] = 최대값이 위치하는 열의 번호
- 예제 :
 - 2 by 3 행렬의 값이 아래와 같을 때, MaxArray 함수는 result에 {6,0,1}을 기록한다.

0행 : 1 6 5
1행 : 2 4 3

output

```
만들고자 하는 2차원 배열의 행과 열의 개수 입력 : 2 3
0행의 배열 요소 3개를 입력하시오 : 1 6 5
1행의 배열 요소 3개를 입력하시오 : 2 4 3

2차원 배열의 최대값은 6
최대값의 위치는 0, 1
계속하려면 아무 키나 누르십시오 . . .
```



• Exam02_02.cpp 파일

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

void MaxArray(int **pArr, int row, int col, int *result);

void main()
{
    int ** pArr;
    int row, col;
    int resultArray[3];

    cout << "만들고자 하는 2차원 배열의 행과 열의 개수 입력 : ";
    cin >> row >> col;

    // pArr에 행의 개수 크기의 일차원 배열 할당
    // 각 행을 표현하기 위한 배열(열의 개수 크기를 가짐)들을 할당

    for(i = 0; i < row; i++)
    {
        // 행렬의 요소를 입력
    }

    MaxArray(_____);

    // 제시한 출력 형식에 맞게 결과 배열을 출력
    // 동적 할당한 메모리 해제
}
```

- 다음의 요구사항에 맞춰 Cart 클래스를 설계하고 구현하라.

목표: 쇼핑카트 클래스를 구현하고, 물건을 넣고 빼며 현재 카트의 물건리스트와 금액을 출력한다.

1. 물건의 ID(int pID)와 금액(int price)을 갖는 PRODUCT 구조체(Struct)를 선언.
2. **클래스 멤버변수**
 1. 10개의 물건 ID를 저장하는 int형 productID 배열
 2. 저장된 물건개수를 나타내는 int nProducts
3. **클래스 멤버함수**
 1. 기본생성자: Cart();
 2. 복사생성자: Cart(const Cart& ca);
 3. 카트에 물건을 넣는 함수: add(int pID)
 4. 카트에서 물건을 빼는 함수: remove(int pID);
 5. 카트의 물건 리스트를 출력하는 함수 print();
 6. 카트의 물건 리스트와 금액을 출력하는 함수 print(PRODUCT productPriceAry[]);

- 다음의 요구사항에 맞춰 Cart 클래스를 설계하고 구현하라.

기타 요구사항

1. 모든 물건 리스트(productPriceAry[])는 다음과 같이 미리 주어짐
 1. {pID, Price} = {1001, 1000}, {2001, 5000}, {2002, 7500}, {3001, 5000}, {3002, 10000}, {3003, 3000}
 2. main()에서 이 물건 리스트를 선언하고 입력함
2. 물건의 추가(Add)와 삭제(Remove)는 위 리스트에 있는 pID를 이용함
Ex) Cart ca; ca.Add(1001);
위와 같이 코드를 작성하면, 1001번 ID의 물건이 카트에 추가됨.
3. Add()시 10개보다 많이 삽입되면 에러 메시지 출력
4. print(PRODUCT productPriceAry[]) 함수의 출력
 1. 물건 ID와 가격을 쌍으로 출력함
 2. print하는 클래스 인스턴스의 pID에 해당하는 금액을 리스트에서 검색하여 함께 출력

• Exam02_03.cpp의 main()

```
int main()
{
    // 물건 리스트(productPriceAry) 선언 및 값 입력 코드작성
    cout << "[Cart management program]" << endl;
    Cart ca1;
    ca1.add(1001);
    ca1.add(2001);
    ca1.print();

    Cart ca2(ca1);
    ca2.remove(1001);
    ca2.add(2002);
    ca2.print(productPriceAry);

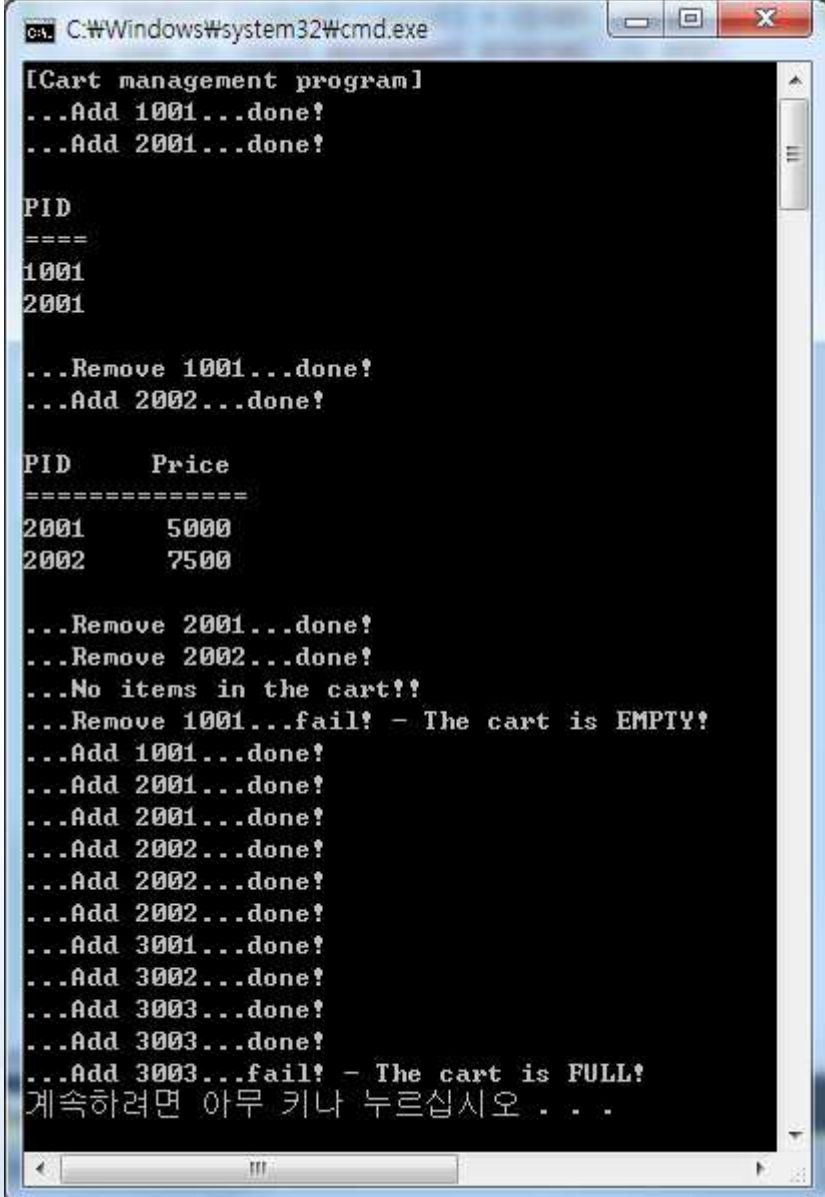
    ca1.remove(2001);
    ca1.remove(2002);
    ca1.print();

    ca1.remove(1001);

    Cart ca3;
    ca3.add(1001);
    ca3.add(2001);
    ca3.add(2001);
    ca3.add(2002);
    ca3.add(2002);
    ca3.add(2002);
    ca3.add(2002);
    ca3.add(3001);
    ca3.add(3002);
    ca3.add(3003);
    ca3.add(3003);
    ca3.add(3003);

    return 0;
}
```

[main() 소스 및 실행결과]



```
C:\Windows\system32\cmd.exe
[Cart management program]
...Add 1001...done!
...Add 2001...done!

PID
==
1001
2001

...Remove 1001...done!
...Add 2002...done!

PID    Price
=====
2001    5000
2002    7500

...Remove 2001...done!
...Remove 2002...done!
...No items in the cart!?
...Remove 1001...fail! - The cart is EMPTY!?
...Add 1001...done!
...Add 2001...done!
...Add 2001...done!
...Add 2002...done!
...Add 2002...done!
...Add 2002...done!
...Add 2002...done!
...Add 3001...done!
...Add 3002...done!
...Add 3003...done!
...Add 3003...done!
...Add 3003...fail! - The cart is FULL!?
계속하려면 아무 키나 누르십시오 . . .
```


● 문제 3의 Cart 클래스를 다음의 요구사항에 맞춰 수정하여라.

목표: 기 구현된 쇼핑카트 클래스의 출력 내용은 동일하되 내부 구현을 다음의 요구사항에 맞춰 수정하여라.

1. 클래스 멤버변수 및 멤버함수

1. 여기에서 언급되지 않아도 원래 함수가 다르게 구현될 수 있음. 예를 들면, 복사생성자는 동적 배열을 할당하도록 다르게 구현되어야 함
2. 카트가 담을 수 있는 물건의 개수를 객체가 생성될 때 정할 수 있게 함. 즉 정적 배열이 아닌 동적 배열을 사용하여 리스트 관리
3. 기본생성자: `Cart();` // 최대 적재수는 기본 10개로 초기화
4. 최대 적재수를 파라미터로 받는 생성자: `Cart(int num);`
5. 아래의 함수에 대해 멤버함수를 전역함수로 대체하여 구현
 1. 카트에 물건을 넣는 함수: `add(Cart& ca, int pID);`
 2. 카트에서 물건을 빼는 함수: `remove(Cart& ca, int pID);`
6. `print()`는 파라미터 없음, 하지만 물건 리스트와 금액을 출력함

- 문제 3의 Cart 클래스를 다음의 요구사항에 맞춰 수정하여라.

기타 요구사항

1. 문제 3의 멤버함수를 외부함수로 구현 시 요구사항
 1. ManagementFunctions.cpp 라는 파일에 Add와 Remove함수 구현
 2. Cart 클래스의 해당 함수를 friend로 선언
2. 물건 아이디와 금액을 저장하는 productPriceAry는 Cart 클래스의 static const 멤버로 선언하고 Cart.cpp에서 초기화함
3. 초기화할 데이터는 문제 3과 동일

• Exam02_04.cpp의 main()

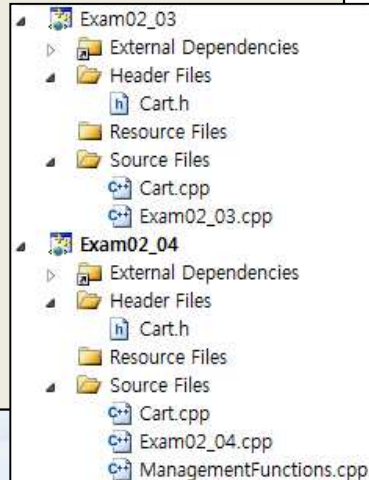
```
int main()
{
    cout << "[Cart management program]" << endl;
    Cart ca1;
    add(ca1, 1001);
    add(ca1, 2001);
    ca1.print();

    Cart ca2(ca1);
    remove(ca2, 1001);
    add(ca2, 2002);
    ca2.print();

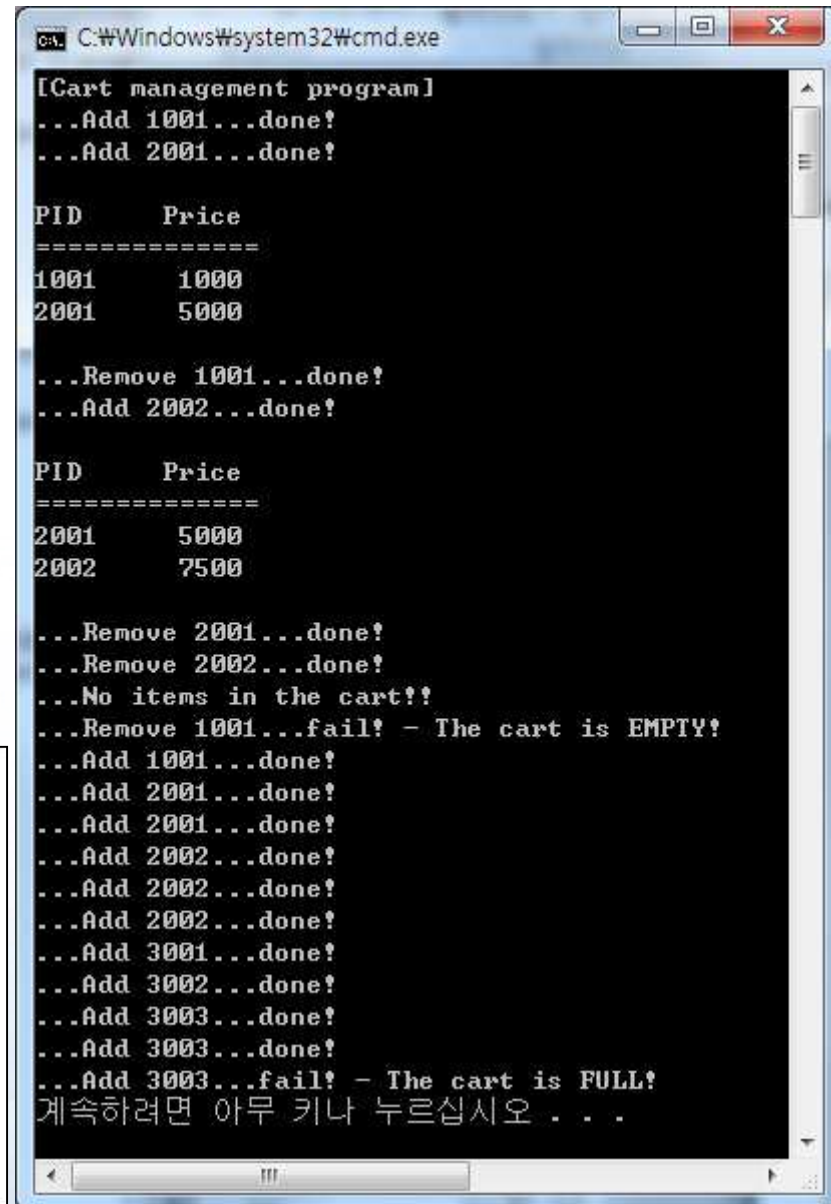
    remove(ca1, 2001);
    remove(ca1, 2002);
    ca1.print();

    remove(ca1, 1001);

    Cart ca3;
    add(ca3, 1001);
    add(ca3, 2001);
    add(ca3, 2001);
    add(ca3, 2002);
    add(ca3, 2002);
    add(ca3, 2002);
    add(ca3, 3001);
    add(ca3, 3002);
    add(ca3, 3003);
    add(ca3, 3003);
    add(ca3, 3003);
    return 0;
}
```



[main() 소스 및 프로젝트 구성]



```
C:\Windows\system32\cmd.exe

[Cart management program]
...Add 1001...done!
...Add 2001...done!

PID      Price
=====
1001      1000
2001      5000

...Remove 1001...done!
...Add 2002...done!

PID      Price
=====
2001      5000
2002      7500

...Remove 2001...done!
...Remove 2002...done!
...No items in the cart!!
...Remove 1001...fail! - The cart is EMPTY!
...Add 1001...done!
...Add 2001...done!
...Add 2001...done!
...Add 2002...done!
...Add 2002...done!
...Add 2002...done!
...Add 3001...done!
...Add 3002...done!
...Add 3003...done!
...Add 3003...done!
...Add 3003...fail! - The cart is FULL!
계속하려면 아무 키나 누르십시오 . . .
```

• 주어진 조건에 맞도록 BankAccount 클래스를 작성하시오

은행 계좌를 정의하는 BankAccount 클래스를 아래의 조건에 맞도록 작성한다.

1. 멤버변수

- int m_nAccountNo ; 계좌번호(5자리 임의의 숫자)
- int m_nBalance ; 예금 잔액
- TRANSACTION m_stHistory[MAX_HISTORY_SIZE]; 최대 MAX_HISTORY_SIZE 개의 거래 내역을 기록
하나의 거래 내역을 변동 금액(정수형)과 적요(최대 50바이트의 문자열)를 멤버로 갖는 TRANSACTION
구조체로 정의함.
- MAX_HISTORY_SIZE는 static const 멤버로 20으로 초기화해야 함.
- static int nTotalBankBalance ; 은행에서 발생한 전체 거래 잔고 (객체중 입금발생시 → 증가, 출금발생시 → 감소)
- int m_nHistoryCount ; 최근 거래 내역의 개수를 저장하는 변수

2. 멤버함수

- BankAccount() : 기본 생성자 - **inline 함수로 구현**
계좌번호를 할당(rand()이용)하고, 나머지 멤버를 모두 0으로 초기화
- BankAccount(int) : 계좌번호를 할당하고, 예금을 전달인자로 입력 받아 이를 적용. 초기 예금으로 **m_nBalance는 초기화 리스트를 이용하여 초기화**
- PrintAccount() : 계좌의 현재 상태를 화면에 출력(결과화면 참조)
- static void PrintBank() ; 은행의 전체 거래 잔고를 화면에 출력(결과화면 참조)
- Deposit(int nAmount, char * strMemo) ; 입금액과 비고를 전달하여, 최근거래 내역과 예금 잔액에 적용
- Withdraw((int nAmount, char * strMemo) ; 출금액과 비고를 전달하여, 최근 거래 내역과 예금 잔액에 적용
- Deposit()과 Withdraw()에서 거래 내역이 MAX_HISTORY_SIZE를 초과하면 에러 출력.

• 주어진 조건에 맞도록 BankAccount 클래스를 작성하시오

```
void BankAccount::Deposit( int nAmount, char * strMemo )  
{  
    if( m_nHistoryCount == MAX_HISTORY_SIZE )  
    {  
        cout << "ERROR!!! 거래 내역이 초과되었습니다 !!! " << endl ;  
        return ;  
    }  
    BankTotalBalance += nAmount ;  
    m_nBalance += nAmount ;  
    m_stHistory[m_nHistoryCount].nAmount = nAmount ;  
    strcpy( m_stHistory[m_nHistoryCount].strMemo, strMemo ) ;  
    m_nHistoryCount++ ;  
}
```

문자열 복사 함수(strcpy())의 사용

- 함수 원형 : char * strcpy(char *, char *) ;
 - 함수 의미 : 문자열을 복사한다.
 - 예제 : strcpy(pDest, pSource) ;
 - pSource 문자열을 pDest에 복사
- ```
char * pStr1 = new char[10] ;
char * pStr2 = "Hello!!" ;
strcpy(pStr1, pStr2) ;
pStr1에 "Hello!!" 문자열이 복사됨
```

### 난수 발생 함수(rand())의 사용

- 헤더파일 : stdlib.h
- 함수 원형 : int rand() ;
- 함수 의미 : 함수 호출시 난수를 반환한다.
- 예제 : int random = rand() ;
  - 정수형 변수 random에 난수를 발생시켜 그 값을 대입한다.

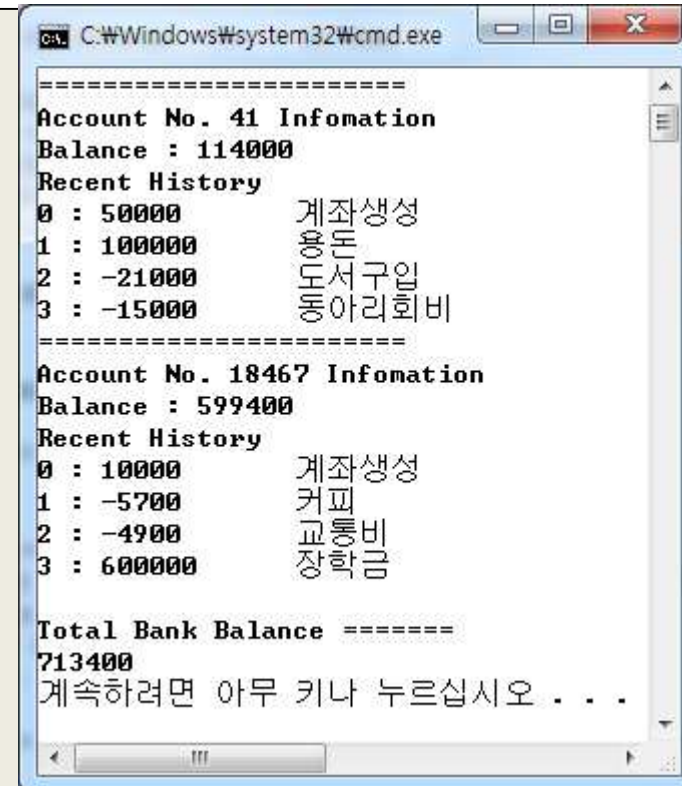
## • Exam02\_05cpp 파일

output

```
#include <iostream>
using namespace std;
int main()
{
 BankAccount A(50000) ;
 A.Deposit(100000, "용돈") ;
 A.Withdraw(21000, "도서구입") ;
 A.Withdraw(15000, "동아리회비") ;
 A.PrintAccount() ;

 BankAccount B(10000) ;
 B.Withdraw(5700, "커피") ;
 B.Withdraw(4900, "교통비") ;
 B.Deposit(600000, "장학금") ;
 B.PrintAccount() ;

 // Bank의 전체 잔고를 출력하는 코드 삽입(객체 A와 B를 사용하지 않는 방법
 // 사용)
 return 0;
}
```



```
=====
Account No. 41 Infomation
Balance : 114000
Recent History
0 : 50000 계좌생성
1 : 100000 용돈
2 : -21000 도서구입
3 : -15000 동아리회비
=====
Account No. 18467 Infomation
Balance : 599400
Recent History
0 : 10000 계좌생성
1 : -5700 커피
2 : -4900 교통비
3 : 600000 장학금
=====
Total Bank Balance =====
713400
계속하려면 아무 키나 누르십시오 . . .
```



## • 주어진 조건에 맞도록 DataType클래스를 작성하시오

여러 종류의 자료형을 저장할 수 있는 DataType 클래스를 구성하시오.

멤버변수

- int m\_nType; 저장된 변수의 타입을 지정하는 변수
- DATA m\_Data; 다음 변수들을 공용체(union)로 정의하여 사용
  - char \* pStr; → 문자열에 대한 포인터
  - double fFloat; → 실수를 저장하는 멤버 변수
  - int nInt; → 정수를 저장하는 멤버 변수

열거형

- NONE, STRING, FLOAT, INTEGER 를 정의하여 m\_nType 멤버 변수에 사용
  - NONE : 데이터의 타입이 정의되어 있지 않음
  - STRING : 데이터의 타입이 문자열임을 나타냄
  - FLOAT : 데이터의 타입이 실수형임을 나타냄
  - INTEGER : 데이터의 타입이 정수형임을 나타냄



## • 주어진 조건에 맞도록 DataType클래스를 작성하시오

### 멤버함수

- DataType()  
m\_nType→NONE으로 설정
- DataType(DataType &)  
객체의 멤버 변수들을 전달 받은 객체의 멤버들로 복사
- DataType(char \*)  
m\_nType→STRING, m\_pString→전달받은 문자열로 복사
- DataType( double )  
m\_nType→FLOAT, m\_fFloat→전달받은 실수형으로 값을 대입
- DataType( int )  
m\_nType→INTEGER, m\_nInt→전달받은 정수형으로 값을 대입
- ~ DataType()  
소멸자 m\_nType이 STRING이라면, m\_pString에 동적할당된 메모리를 반환
- void PrintData()  
m\_nType을 검사하여 저장된 변수의 데이터 값을 화면에 출력

## • 주어진 조건에 맞도록 DataType클래스를 작성하시오

### 연산자 오버로딩

- + 연산자 오버로딩 ( DataType 객체 + DataType 객체 )  
두 DataType 객체의 데이터를 서로 더한 DataType 객체를 반환(문자열일 경우에는 두 문자열을 붙여넣음)  
DataType의 m\_nType을 검사하여, 서로 데이터 형이 다르다면 에러를 출력하며 기본 생성자로 객체를 만들어서(또는 이용해서)를 반환
- + 연산자 오버로딩 ( char \* + DataType객체 )  
STRING에 대해서는 main()의 StringD, StringE와 같이 DataType객체와의 덧셈을 허용할 수 있는 +연산자 오버로딩을 friend를 이용한 외부함수로 구현.
- = 연산자 오버로딩 ( DataType 객체 = DataType 객체 )  
우측에 있는 DataType 객체의 데이터를 좌측의 DataType 객체에 대입

### 주의

- STRING을 저장하기 위한 char 배열을 동적 할당할 때 "문자열의 길이 + 1" 크기만큼 할당해야 함.
- 추가된 한 바이트는 문자열의 끝을 나타내는 NULL 문자('W0')를 저장하기 위해 사용됨
- Operator+를 오버로딩할 때는 "문자열1의 길이 + 문자열2의 길이 + 1" 크기만큼 할당

### 문자열 복사 함수(strlen())의 사용

- 함수 원형 : int \* strlen( char \* );
- 함수 의미 : 문자열을 길이를 반환.
- 예제 : int len = strlen( pStr );  
 • pStr 문자열에서 NULL문자('W0')를 제외한 문자열의 길이를 반환한다.  
 char \* pStr = "ABEDF";  
 int nLen = strlen(pStr);  
 nLen → 5의 값이 대입 됨

### 문자열 결합 함수(strcat())의 사용

- 함수 원형 : char \* strcat( char \*, char \* );
- 함수 의미 : 문자열을 결합한다.
- 예제 : strcat( pFirst, pSecond );  
 • pFirst 문자열의 뒤에 pSecond 문자열을 결합  
 char \* pFirst = "AB"  
 char \* pSecond = "EDF"  
 strcat( pFirst, pSecond); -> pFirst="ABEDF";

## • Exam02\_06.cpp 파일

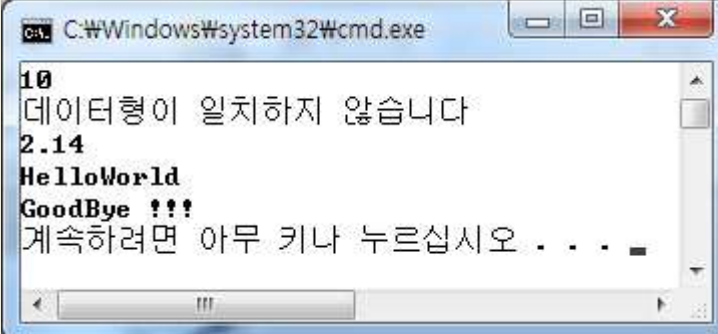
output

```
#include <iostream>
using namespace std;
int main()
{
 DataType IntA(55);
 DataType IntB(-45);
 DataType IntC ;
 IntC = IntA + IntB ;
 IntC.PrintData() ;

 DataType FloatA(3.14);
 DataType FloatB(-1.0);
 DataType FloatC ;
 FloatC = IntA + FloatB ; // 데이터 타입이 일치하지 않아 화면에 에러를 출력
 FloatC = FloatA + FloatB ;
 FloatC.PrintData() ;

 DataType StringA("Hello");
 DataType StringB("World");
 DataType StringC ;
 StringC = StringA + StringB ;
 StringC.PrintData() ;

 DataType StringD("Bye !!!");
 DataType StringE ;
 StringE = "Good" + StringD ;
 StringE.PrintData() ;
 return 0;
}
```



```
C:\Windows\system32\cmd.exe
10
데이터형이 일치하지 않습니다
2.14
HelloWorld
GoodBye !!!
계속하려면 아무 키나 누르십시오 . . .
```