

# Exam #1



경희대학교  
KYUNG HEE UNIVERSITY

## - 주의 사항

- 참고할 자료들을 미리 다운로드 받은 후, 시험 시작 5분 전 네트워크 차단
- 스마트 폰, 전자사전 등의 전자기기 일절 사용 금지
- 시험 시작 후 출입 불가

## - 업 로드 방법

- 실습/과제와 동일한 방법으로 솔루션 구성(LAB Help 자료 참고)
- <http://dke.khu.ac.kr/aoop> 의 [Assignments Submission] 게시판에 업 로드
- 파일명과 게시물 제목을 아래와 같이 통일
  - [분반]\_ex01\_학번
  - Ex) [A]\_ex01\_2012345678

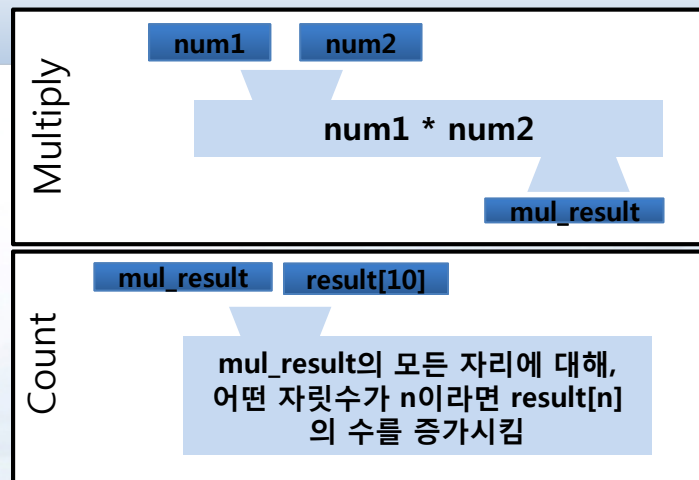
- 두 양의 정수를 곱하여 나온 양의 정수에서 0-9의 숫자가 나타나는 개수를 측정하는 프로그램을 작성하시오.

목표: 사용자로부터 양의 정수 2개를 입력 받아 곱을 계산하는 함수를 작성하여 곱을 계산하고, 곱셈의 결과값에서 0-9가 몇 번씩 나타나는지 세는 함수를 작성한다.

1. 사용자로부터 양의 정수 2개를 입력 받는다.
2. Multiply 함수를 호출하여 두 개의 정수의 곱을 계산 (이 때, 곱셈의 결과값은 1부터  $2^{32} - 1$  사이에 있다고 가정함)
3. Count 함수를 호출하여 곱셈의 결과값에서 0-9가 몇 번씩 나타나는지 세어 배열에 저장 (Count함수의 인자값으로 Multiply의 반환값과 크기 10의 배열을 전달)

\* 크기 10의 배열은 int형이고 크기가 10이며, result[n]은 n이 곱의 결과에서 나타난 횟수를 저장한다

4. 배열을 출력



### 결과출력

```

첫 번째 수를 입력하시오.2932
두 번째 수를 입력하시오.1092
곱의 결과 :3201744
  
```

```

0<이>가 나타난 횟수 : 1
1<이>가 나타난 횟수 : 1
2<이>가 나타난 횟수 : 1
3<이>가 나타난 횟수 : 1
4<이>가 나타난 횟수 : 2
5<이>가 나타난 횟수 : 0
6<이>가 나타난 횟수 : 0
7<이>가 나타난 횟수 : 1
8<이>가 나타난 횟수 : 0
9<이>가 나타난 횟수 : 0
  
```

## • Main.cpp 파일

```
int main()
{
    int a, b; // 입력받을 양의 정수 2개가 저장될 변수
    // 1. 곱의 결과가 저장될 변수를 선언 (mul_result)
    int i;
    int result[10];
    // 2. 사용자로부터 2개의 양의 정수를 입력받음

    // 3. Multiply 함수를 호출
    cout << "곱의 결과 :" << mul_result<<endl;

    // 4. Count 함수를 호출

    // 5. 결과 배열을 출력

    return 0;
}
```

## • 다음 조건을 만족하도록 프로그램을 작성하라.

목표: 이차 방정식의 해를 구하는 프로그램을 작성하라.

- 조건 :

1. 구하고자 하는 이차 방정식은  $ax^2+bx+c = 0$  형태의 이차 방정식이다.
2. 구현 목록의 2개의 함수를 구현한다.
3. 2에서 구현한 함수를 이용해서 실수형 계수  $a, b, c$ 를 사용자로 부터 입력 받는다. 이때  $a$ 의 값이 반드시 0이 아닌 값을 입력 받는다고 가정한다.
4. 2에서 구현한 함수와 3에서 입력 받은 변수를 이용해서 이차 방정식의 해를 구한다.
5. 이차 방정식의 해의 개수(2개, 1개, 없다)를 화면에 출력해 주고 해의 값을 화면에 출력해 준다.
6. 결과출력의 화면과 같이 실행되도록 한다.

**결과출력**

### GetCoefficients 함수

- 함수 원형 : void GetCoefficients (float &a, float &b, float &c);
- 기능 : 3개의 정수를 키보드로 입력 받아  $a, b, c$  변수에 저장한다.

### QuadEqSlove 함수

- 함수 원형 : int QuadEqSlov(float pA, float pB, float pC, float\* pX1, float\* pX2);
- 기능 : 3개의 매개변수를 계수로 가지는 이차 방정식의 해를 구해 pX1과 pX2포인터가 가리키는 실수에 저장. 해가 하나일 때는 두 포인터가 가리키는 실수에 같은 값을 저장하고 해가 없을 때는 0을 저장한다. 오른쪽의 근의 공식과 판단식을 이용한다. 제곱근은 sqrt()함수를 이용해서 구한다.
- 반환값 : 해가 두 개이면 2를 반환 하나면 1을 없으면 0을 반환 한다.

근의 공식

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

판단식

$$D = b^2 - 4ac$$

$D > 0$ , 해가 2개  
 $D = 0$ , 해가 1개  
 $D < 0$ , 실수해 없음

이차 방정식 계수 a :10  
 이차 방정식 계수 b :1  
 이차 방정식 계수 c :1  
 해가 없습니다.

이차 방정식 계수 a :4  
 이차 방정식 계수 b :4  
 이차 방정식 계수 c :1  
 하나의 해를 구했습니다.  
 1 번째 해는 -0.5

이차 방정식 계수 a :2  
 이차 방정식 계수 b :4  
 이차 방정식 계수 c :1  
 두개의 해를 구했습니다.  
 1 번째 해는 -0.292893  
 2 번째 해는 -1.70711

## • 부분 단조증가수열들을 찾아 순서를 바꾸는 프로그램을 작성하라.

주어진 수열에서 길이가 2이상인 모든 부분 단조증가수열들을 찾아 그 수열에 속한 수들의 순서를 역으로 바꾸는 프로그램을 아래의 조건에 맞게 작성하라.

1. 입력은 두 줄로 구성된다. 첫 번째 줄은 수열에 있는 수의 개수이고, 두 번째 줄은 공백 문자로 구분된 수열 값들이다.
2. 입력으로 주어진 수열 크기에 맞게 배열을 동적 할당하여 수열 값들을 읽어들인다. (2점)
3. 아래에서 제시한 두 개의 함수 findFirstSequence와 reverseOrder를 구현하고, 이를 사용하여 실행 예와 같이 되도록 프로그램을 작성하시오. (reverseOrder 2점, findFirstSequence 4점, main 2점)
4. 힌트: findFirstSequence의 반환값이 0이 될 때까지 (파라미터를 적당히 바꿔가며) 반복해서 findFirstSequence를 호출. 매 호출 후에 reverseOrder를 호출하여 해당 부분 수열의 순서를 변경함.

### findFirstSequence 함수

- **파라미터:**
  - sequence: 수열 값을 저장하고 있는 배열의 시작 주소
  - seq\_size: 수열에 있는 원소의 개수
  - result\_seq\_ptr: 찾은 부분 단조증가수열의 첫 번째 원소 주소
- **기능 :** sequence에서 2개 이상의 원소로 구성된 첫 번째 부분 단조증가수열을 찾아 그 첫 번째 원소의 주소를 result\_seq\_ptr를 통해 반환한다.
- **반환값 :** 찾은 부분 단조증가수열에 있는 원소의 개수. 못 찾은 경우 0을 반환.

### reverseOrder 함수

- **파라미터:**
  - sequence: 수열 값을 저장하고 있는 배열의 시작 주소
  - seq\_size: 수열에 있는 원소의 개수
- **기능 :** sequence에 있는 원소의 값을 역순으로 변경함. 예를 들면, (3, 7, 9, 11) 이 입력 수열이면 (11, 9, 7, 3)의 순으로 변경함.
- **반환값 :** 없음.

### 부분 단조증가수열

- 단조증가수열이란 수열  $a_1, a_2, a_3, \dots, a_n, a_{n+1}, \dots$ 이 조건  $a_1 < a_2 < a_3 < \dots < a_n < a_{n+1} < \dots$ 을 만족하는 수열이다.
- 부분 단조증가수열이란 수열 내에서 단조증가수열 조건을 만족하는 최대 연속 부분이다.
- 예: 수열 (3, 7, 9, 11, 9, 5, 4, 15, 10, 6, 14, 18)에서 부분 단조증가수열은 (3, 7, 9, 11), (9), (5), (4, 15), (10), (6, 14, 18)의 총 여섯 개이다. 이 중에서 길이가 2이상인 것은 (3, 7, 9, 11), (4, 15), (6, 14, 18)의 총 세 개이다

### 실행 예

```
수열의 크기: 12
수열: 3 7 9 11 9 5 4 15 10 6 14 18
변환된 수열: 11 9 7 3 9 5 15 4 10 18 14 6
계속하려면 아무 키나 누르십시오 . . .
```



## ● 정수 행렬을 표현하기 위한 MaxtrixInt 클래스를 구현하라.

목표: 다음 페이지에 주어진 main 함수를 분석하고 이와 같이 동작할 수 있게 MatrixInt 클래스를 구현한다.

1. 아래 프로젝트구조와 같이 main.cpp 파일과 MatrixInt.h 헤더파일, MatrixInt.cpp 구현파일을 만들어라.
2. 아래와 같이 결과가 출력되도록 MatrixInt 클래스를 구현하라, 이때 정보은닉을 반드시 고려해서 클래스를 구현한다.
3. 매트릭스의 값은 int 형 2차원 정적 배열에 저장하고 최대 사이즈는 10x10 이 되도록 한다. 이때 기본 생성자는 2x2 행렬을 생성한다.
4. 이때 매트릭스 값을 int 형 2차원 정적 배열이 아닌 int\*\*로 선언하고 아래 그림과 같은 방법으로 행렬 사이즈 만큼 동적으로 배열을 만드는 클래스를 작성하고 소멸자에서 동적 할당된 배열을 올바르게 해지 하는 경우 추가 점수 부여.

### 결과출력

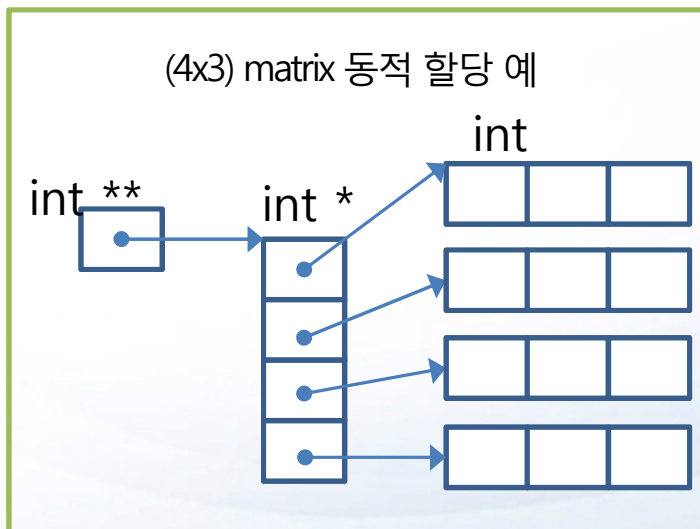
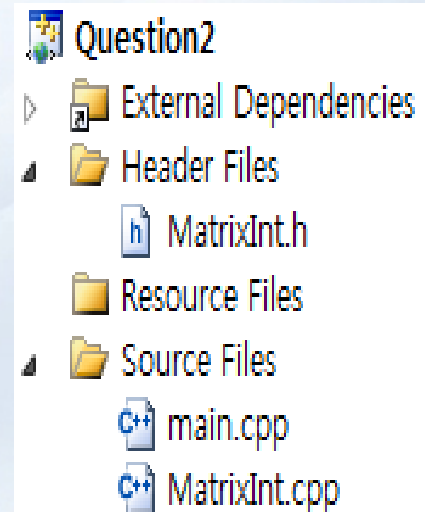
```

Input matrix1
Matrix<0,0> = 1
Matrix<0,1> = 1
Matrix<1,0> = 1
Matrix<1,1> = 1

input matrix2
Matrix<0,0> = 1
Matrix<1,0> = 2
show matrix1
1 1
1 1

show matrix2
1
2
  
```

### 프로젝트구조



## • Main.cpp 파일

```
#include <iostream>
#include "MatrixInt.h"

using namespace std;

int main()
{
    // 매트릭스1 인스턴스를 기본 생성자를 이용하여 생성함
    // 기본 생성자는 2x2매트릭스를 기본으로 생성
    MatrixInt matrix1;

    // 매트릭스 값을 키보드로 부터 입력 받는다.
    cout<<"input matrix1"<<endl;
    matrix1.inputData();

    // 매트릭스2 인스턴스를 아래 생성자를 이용하여 생성함
    // 첫번 파라미터는 행 크기를 두번째 파라미터는 열 크기를
    MatrixInt matrix2(2,1);

    // 매트릭스 값을 키보드로 부터 입력 받는다.
    cout<<endl<<"input matrix2"<<endl;
    matrix2.inputData();

    // matrix1 값 출력
    cout<<endl<<"show matrix1"<<endl;
    matrix1.showMatrix();

    // matrix2 값 출력
    cout<<endl<<"show matrix2"<<endl;
    matrix2.showMatrix();
    return 0;
}
```



### ● 히스토그램을 표현하기 위한 프로그램을 구현하라.

정수들의 집합이 주어졌을 때, 데이터 분포를 나타내는 히스토그램을 구하기 위해 클래스 Histogram을 구현하고 이를 이용하여 프로그램을 작성한다.

1. 히스토그램은 주어진 정수 데이터 집합에 속한 수들의 최소값과 최대값을 양끝으로 하는 범위(데이터 범위라 하자)를 여러 개의 구간으로 나누고, 각 구간에 속한 수의 출현 개수를 나타낸 것이다. 데이터 범위 크기는 구간 개수보다는 크다고 가정한다.
2. 모든 구간은 동일한 크기로서, 데이터 범위 크기를 구간 개수로 나눈 값이다. 데이터 범위 크기가 구간 개수의 배수라고 가정한다.
3. 입력은 세 줄로 구성된다. 첫째 줄은 데이터 집합에 있는 수의 개수, 둘째 줄은 공백 문자로 구분된 데이터들, 셋째 줄은 히스토그램에 사용되는 구간의 개수를 나타낸다.
4. 다음 페이지 프로그램 실행 결과가 아래 실행 예와 같이 되도록, main.cpp의 빠진 부분을 보완하여 작성하고, 클래스 Histogram을 작성하시오.
5. AddHistograms() 함수는 구간 정보가 동일한 두 히스토그램을 입력으로 받아 각 대응 구간의 빈도 수를 더해서 만들어지는 히스토그램을 출력한다.
6. Histogram에서 구간 별 출현 개수를 저장하는 배열을 동적 배열로 구현하시오. 정적 배열로 구현하면 감점.

#### 히스토그램 예:

데이터 집합이 {3, 9, 8, 7, 6, 14, 5, 4, 5, 8, 10, 12}인 경우, 3개의 구간으로 구성된 히스토그램은 다음과 같음.

구간 3 - 6: 5

구간 7 - 10: 5

구간 11 - 14: 2

## • Main.cpp 파일

```
#include "Histogram.h"

int main()
{
    int num_items; // 집합 원소 개수
    int *data_ary; // 집합 원소들을 저장하는 동적 배열의 시작 주소
    int num_intervals; // 히스토그램에 사용되는 구간 개수

    cout << "집합 원소 개수와 집합 원소들을 입력하시오 ." << endl;

    // 입력 데이터를 읽는 코드를 작성하시오.
    // data_ary는 데이터 개수만큼 동적 배열을 할당하시오.

    Histogram h1(data_ary, num_items, num_intervals);
    cout << "### Histogram h1" << endl;
    h1.print();

    Histogram h2(h1); // h1을 복사해서 h2를 만듦
    cout << "### Histogram h2" << endl;
    h2.print();

    cout << "### Histogram h1+h2" << endl;
    AddHistograms(h1, h2); // h1 + h2를 출력

    return 0;
}
```

## 실행 예

```
집합 원소 개수와 집합 원소들을 입력하시오.
12
3 9 8 7 6 14 5 4 5 8 10 12
3
### Histogram h1
3 - 6: 5
7 - 10: 5
11 - 14: 2
### Histogram h2
3 - 6: 5
7 - 10: 5
11 - 14: 2
### Histogram h1+h2
3 - 6: 10
7 - 10: 10
11 - 14: 4
계속하려면 아무 키나 누르십시오 . . .
```