

Exam #1



- 주어진 question1.cpp 파일을 두 집합의 차집합 원소의 개수를 반환하는 함수를 작성하여라.

목표: 두 개의 배열에 원소들을 받고 차집합 원소의 개수를 반환하는 함수를 작성한다.

- 두 배열의 원소 개수와 원소의 값은 사용자로부터 입력 받는다.
- 사용자가 중복 원소를 입력하지 않음을 가정한다.
- 아래 결과와 같이 출력되도록 소스를 작성함

함수 프로토타입

- `int CountSetDifference (int a[], int b[], int a_size, int b_size);`
 - 반환값 : a와 b 배열의 차집합 원소의 개수를 int형 정수로 반환
 - 인자 : int형 원소로 구성되는 두 배열 a, b와 각 집합의 원소 개수를 의미하는 a_size와 b_size를 전달.
 - 기능 : a와 b 배열의 차집합(a - b) 원소의 개수를 센다.

output

```
집합 1의 원소의 개수를 입력하세요. :5
집합 2의 원소의 개수를 입력하세요. :7

집합 1의 원소를 입력하세요. :1 3 5 7 9
집합 2의 원소를 입력하세요. :10 1 5 20 30 50 9

차집합(a-b) 원소의 개수 :2
```

• question1.cpp 파일

```
int main()
{
    int alen, blen;
    int i,j;
    cout << "집합 1의 원소의 개수를 입력하세요. :";
    // 1번째 집합의 원소의 개수를 입력 받음
    cout << "집합 2의 원소의 개수를 입력하세요. :";
    // 2번째 집합의 원소의 개수를 입력 받음

    cout << "\n";
    cout << "집합 1의 원소를 입력하세요. :";
    // 사용자로부터 집합 1의 원소를 입력받음

    cout << "집합 2의 원소를 입력하세요. :";
    // 사용자로부터 집합 2의 원소를 입력받음

    cout << "\n";
    cout << "차집합 원소의 개수 :";
    cout << CountSetDifferene( ) << endl;

    return 0;
}
```

- 주어진 세 정수형 변수 중 중간 값의 주소를 반환하는 함수를 작성하시오.

세 정수형 변수를 매개변수로 전달 받아 이에 대한 값들 중 중간 값의 주소를 반환하는 findMiddle() 함수를 작성하시오.

```
int main( void )  
{
```

```
    int * pMiddle ;
```

```
    int a = 3 ;
```

```
    int b = 5 ;
```

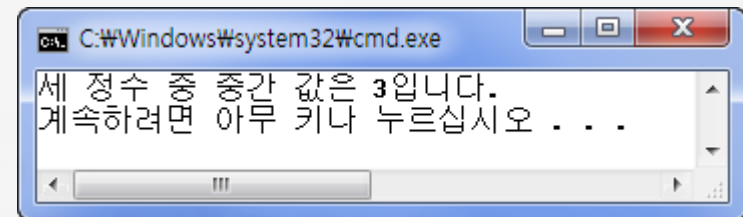
```
    int c = 1 ;
```

```
    pMiddle = findMiddle( &a, &b, &c ) ;
```

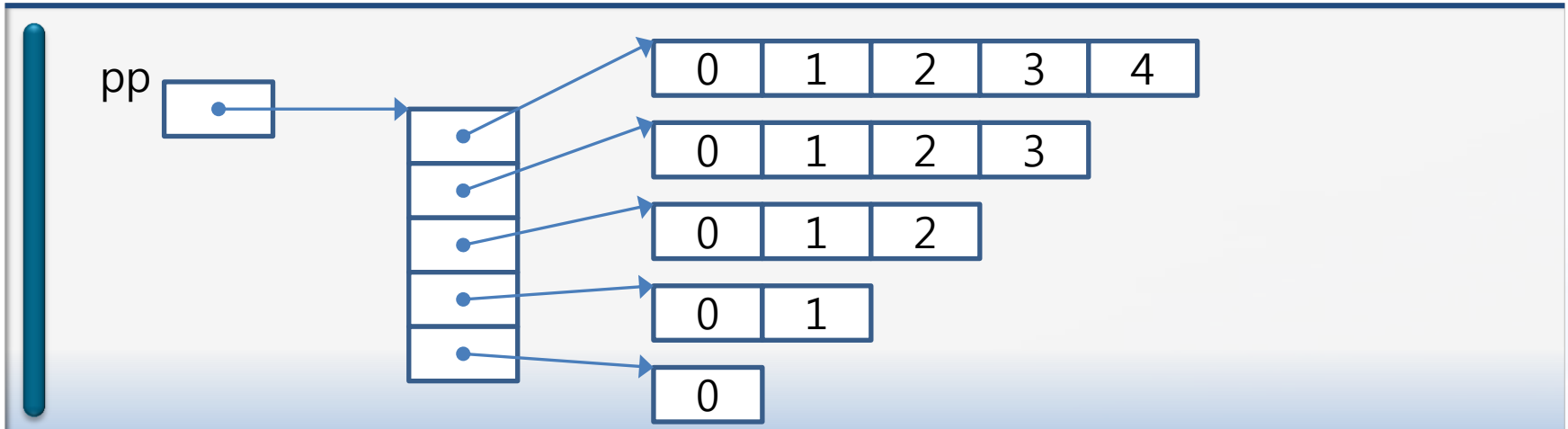
```
    cout << "세 정수 중 중간 값은 " << *pMiddle << "입니다." << endl ;  
    return 0 ;
```

```
}
```

output



- 이중 포인터를 이용하여 아래와 같이 메모리를 할당하고, 이를 초기화 하시오.



- 이중 포인터를 통한 메모리 할당
 - nCount : row 개수
 - column 개수는 밑은 1에서 위로 1씩 증가
- 할당한 메모리에 대한 값 초기화
- output과 같이 값을 화면에 출력
- 선언된 변수 이외 추가 변수 사용 불가
- nCount 값을 주어진 값으로 가정하지 말 것. 값을 변경하여 실행해도 동작하여야 함

output

```

C:\Windows\system32\cmd.exe
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0
계속하려면 아무 키나 누르십시오 . . .
  
```

• question3.cpp 파일

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

int main()
{
    int nCount = 5 ;           // 1차로 할당될 원소의 개수, 이 값은 바뀔 수 있음

    int ** pp ;               // 2차원 포인터를 이용
    int row, col;             // for-loop 를 위한 변수
```

코드 작성

```
        return 0 ;
    }
```

- 주어진 question4.cpp 파일을 보고 Employee 클래스를 생성하여라.

목표: 주어진 소스를 분석하여 Employee 클래스를 생성한다.

1. Employee.h 헤더파일과 Employee.cpp 구현파일을 각각 생성함
2. 아래 결과와 같이 출력되도록 소스를 작성함

output

[첫째 사원 정보]

사원 이름 : 유재석

사원 번호 : 33170

월 급여액 : 5000000

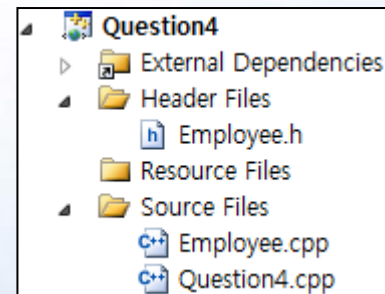
[둘째 사원 정보]

사원 이름 : 박명수

사원 번호 : 31092

월 급여액 : 1000000

프로젝트 구조



• question4.cpp 파일

```
#include <iostream>
#include "Employee.h";
using namespace std;

int main()
{
    /* 사원(employee) emp1 인스턴스를 생성함 */
    Employee emp1;

    /* 생성한 인스턴스를 사원 정보(이름, 사번, 월급여액)를 입력하여 초기화함 */
    emp1.initialize("유재석", 33170, 5000000);

    /* 사원(employee) emp2 인스턴스를 생성함 */
    Employee emp2("박명수", 31092, 1000000);

    /* 첫번째 사원의 정보를 출력함 */
    cout << "[첫째 사원 정보]" << endl;
    emp1.show_data();

    /* 두번째 사원의 정보를 출력함 */
    cout << "[둘째 사원 정보]" << endl;
    emp2.show_data();

    /* emp1과 emp2 객체 생성 방법의 차이를 기술하시오. */
    cout << "여기에 답을 쓰시오." << endl;
    return 0;
}
```


- 주어진 question5.cpp 파일을 보고 계산기 프로그램(Calculator)을 만들어라.

목표: 주어진 소스를 분석하여 Calculator 클래스를 생성한다.

1. Calculator.h 헤더파일과 Calculator.cpp 구현파일을 각각 생성함
2. 아래 결과와 같이 출력되도록 소스를 작성함

output

\$\$ 누구나 만드는 계산기 프로그램(종료: 0을 입력) \$\$

연산 기호를 입력하세요(+,-,*,/,O) : +

첫번째 수를 입력하세요 : 100

두번째 수를 입력하세요 : 75.25

계산결과 : 175.25

연산 기호를 입력하세요(+,-,*,/,O) : -

첫번째 수를 입력하세요 : 27

두번째 수를 입력하세요 : 100

계산결과 : -73

연산 기호를 입력하세요(+,-,*,/,O) : *

첫번째 수를 입력하세요 : 17

두번째 수를 입력하세요 : 25.999

계산결과 : 441.983

연산 기호를 입력하세요(+,-,*,/,O) : /

첫번째 수를 입력하세요 : 16

두번째 수를 입력하세요 : 7

계산결과 : 2.28571

연산 기호를 입력하세요(+,-,*,/,O) : &

ERROR! 연산 기호가 잘못 입력되었습니다!

연산 기호를 입력하세요(+,-,*,/,O) : /

첫번째 수를 입력하세요 : 100

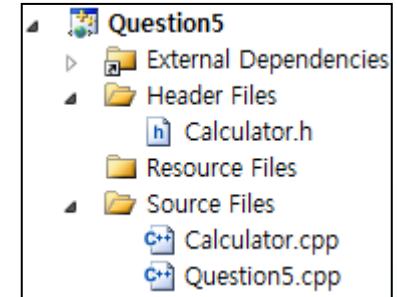
두번째 수를 입력하세요 : 0

ERROR! 0으로 나누는 연산은 불가능합니다!

연산 기호를 입력하세요(+,-,*,/,O) : 0

계속하려면 아무 키나 누르십시오 ...

프로젝트 구조



• question5.cpp 파일

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

#include "Calculator.h"

int main()
{
    float inputNum1 = 0.0;
    float inputNum2 = 0.0;
    char inputOper = '0';
    Calculator cal;

    cout << "$$ 누구나 만드는 계산기 프로그램(종료: 0을 입력) $$" << endl;

    while(1)
    {
        cout << endl << "연산 기호를 입력하세요(+,-,*,/,O) : ";
        cin >> inputOper;

        /* 연산 기호로 0을 넣으면 종료 */
        if( inputOper == '0') break;
        /* 연산 기호가 제대로 입력되었는지 검출 */
        if( cal.operValidator(inputOper) == true )
        {
            /* 제대로 입력되었을 시 계속 진행 */
            cout << "첫번째 수를 입력하세요 : ";
            cin >> inputNum1;
            cout << "두번째 수를 입력하세요 : ";
            cin >> inputNum2;
```

```
/* 나누기 연산이 가능한지 검출 */
if( cal.divValidator(inputNum2, inputOper) == true )
{

    /* 나누기 연산이 가능한 경우라면 주어진 연산을 수행 */
    cout << "계산결과 : ";

    /* 연산을 수행하는 메소드 호출 */
    cal.operation(inputNum1, inputNum2, inputOper);

    /* 연산 결과를 출력하는 메소드 호출 */
    cal.printResult();
}

/* 입력값과 기호가 부적절 하면 다시 입력 받음 */
else
{
    // 아무 작업도 하지 않음
}

/* 입력값과 기호가 부적절 하면 다시 입력 받음 */
else
{
    // 아무 작업도 하지 않음
}
};
return 0;
};
```