

실습문제

1

- 교재 316페이지, 실습문제 6-6
 - ▣ 프로젝트명 : 학번_Chap_6_6
 - ▣ main 파일명 : 학번_Chap_6_6.cpp
 - ▣ 동일한 크기의 배열을 변화하는 다음 2개의 static 멤버 함수를 가진 ArrayUtility2 클래스를 만들고, 이 클래스를 이용하여 교재의 결과(교재 참고)와 같이 출력되도록 프로그램을 완성하라.
 - 입력 배열의 크기는 5로 동일

```
// s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴
static int* concat(int s1[], int s2[], int size);

// s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴
// 배열의 크기는 retSize에 전달. retSize가 0인 경우 nullptr 리턴
static int* remove(int s1[], int s2[], int size, int&retSize);
```

main() 함수

2

```
int main() {
    int 배열 x[5], y[5] 선언
    "정수를 5 개 입력하라. 배열 x에 삽입한다>> " 출력
    for문을 이용하여 각 정수를 배열 원소 x[i]에 입력 받음    // 예제 4-6, 17행 참조

    "정수를 5 개 입력하라. 배열 y에 삽입한다>> " 출력
    for문을 이용하여 각 정수를 배열 원소 y[i]에 입력 받음

    int 포인터 변수 p를 선언함과 동시에 // 예제 5-6, 31행 참조: 변수선언 및 초기화
    ArrayUtility2의 정적(static) 함수 concat(x,y,5)를 호출하고 리턴 값을 p에 저장
    // 클래스명으로 static 멤버 함수 호출은 예제 6-10, 12~14행 참조
    "합친 정수 배열을 출력한다" 와 줄 바꾸기 출력
    for문을 이용하여 각각의 p[i]와 ' '를 출력; 모든 원소 출력 후 줄 바꾸기 출력

    int retSize; // 배열 q의 원소 개수를 저장함
    int 포인터 변수 q를 선언함과 동시에 // 예제 5-6, 31행 참조: 변수선언 및 초기화
    ArrayUtility2의 정적 함수 remove(x,y,5,retSize)를 호출하고 리턴 값을 q에 저장
    "배열 x[]에서 y[]를 뺀 결과를 출력한다. 개수는 ", retSize, 줄 바꾸기 출력
    for문을 이용하여 각각의 q[i]와 ' '를 출력 // 배열 q의 원소 개수?
    줄 바꾸기 출력

    배열 p와 q를 힙(free space 영역)에 반환
}
```

class ArrayUtility2 선언

3

```
class ArrayUtility2 {  
public:  
    // s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴  
    // size는 각 배열의 크기이며 리턴된 배열의 크기는 size*2이다.  
    static int* concat(int s1[], int s2[], int size);  
  
    // s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴  
    // 리턴하는 배열의 크기는 retSize에 저장.  
    // retSize가 0인 경우 nullptr 리턴  
    static int* remove(int s1[], int s2[], int size, int&retSize);  
};
```

ArrayUtility2의 concat() 멤버 함수

4

```
int* ArrayUtility2::concat(int s1[], int s2[], int size) {  
    // 각 배열의 크기는 size  
    // int[] 할당 및 에러 체크는 예제 4-6 참조  
    s1[]의 두 배 크기의 int 배열을 동적으로 할당하여 포인터 변수(선언) p에 저장  
    if 메모리가 할당되지 않았다면, nullptr 리턴  
  
    // s1[]을 p에 복사한 후 그 뒤에 이어서 s2[]를 p에 연속하여 복사한다.  
    for문을 이용하여  
        s1[i]를 p[i]에 복사  
    for문을 이용하여  
        s2[i]를 p[i+?]에 복사  
  
    return p;  
}
```

ArrayUtility2의 remove() 멤버 함수

5

```
int* ArrayUtility2::remove(int s1[], int s2[], int size, int&retSize) {  
    // 각 배열의 크기는 size  
    s1[] 크기의 int 배열을 동적으로 할당하여 포인터 변수(선언) p에 저장  
    if 메모리가 할당되지 않았다면, retSize 값 0으로 설정 후 nullptr 리턴  
    int k=0; // 결과 배열 p의 첨자로 사용할 변수 선언  
    for문을 이용하여 배열 s1의 각각의 요소에 대해 { // for문의 index는 i  
        // 아래는 s1[i] 원소가 배열 s2[]에 존재하는지 체크하여 있으면 break  
        int j; // 아래 for문의 index로 사용할 변수 선언  
        for문을 이용하여 배열 s2의 각각의 요소에 대해 { // index는 j 사용  
            if s1[i]와 s2[j]가 같으면  
                break;  
        } // s1[i] 값이 배열 s2에 존재하면 j는 size보다 작은 값임  
        if (j == size) { // s1[i] 값이 배열 s2에 존재하지 않은 경우임  
            s1[?]의 값을 배열 p[??]에 복사하고, 배열 p의 첨자??를 1 증가 시킴  
        }  
    }  
    } // 뒤 페이지 계속
```

ArrayUtility2의 remove() 멤버 함수-계속

6

// 앞 페이지에 이어서 계속

배열 p에 저장된 s1[]의 원소의 개수를 retSize에 저장
retSize가 0이면 nullptr 리턴

결정된 retSize 크기의 int 배열을 새로 할당 받아 포인터 변수(선언) q에 저장
if 메모리가 할당되지 않았다면, retSize 값 0으로 설정 후 nullptr 리턴

for문을 이용하여 p[i]를 q[i]로 복사(배열 q의 크기만큼만 복사)

이제 더 이상 필요 없는 배열 p를 힙(free space 영역)에 반환

return q;

}