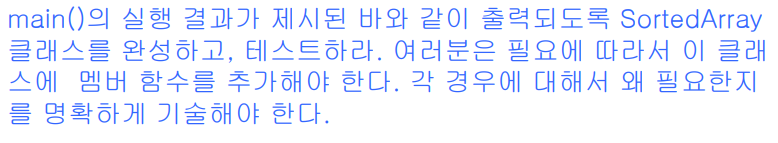


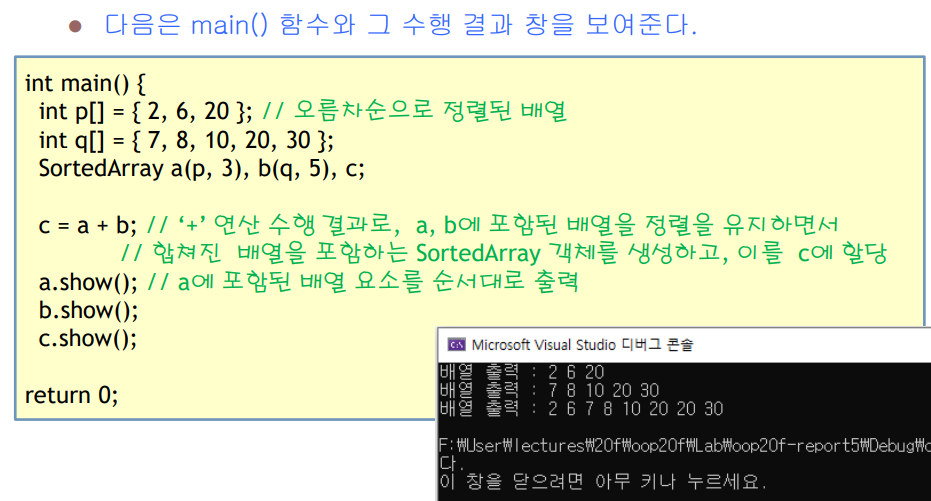
객체지향프로그래밍

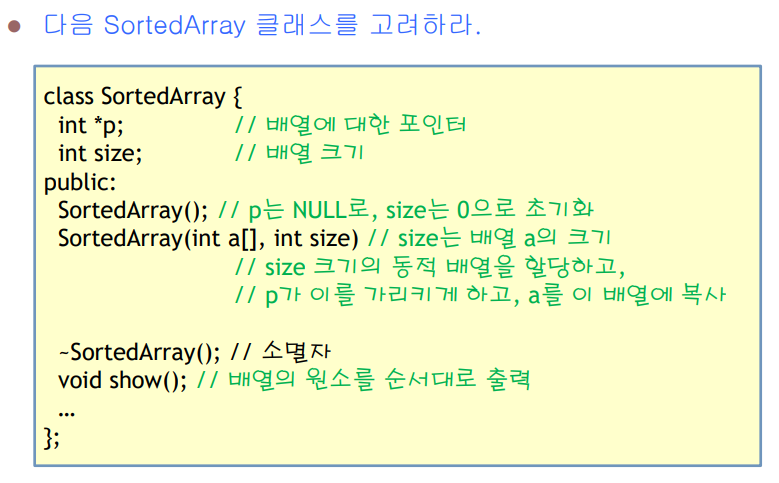
Report #5

|  |  |
| --- | --- |
| 학번 | 20174627 |
| 이름 | 김혜진 |

문제 및 조건



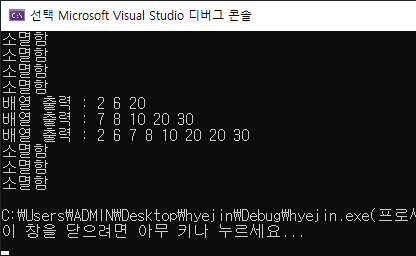




코드 및 설명(주석)

|  |
| --- |
| #include<iostream>  using namespace std;  class SortedArray {  int\* p; // 배열에 대한 포인터  int size; // 배열 크기  public:  SortedArray() {  p = NULL; // p는 NULL로 초기화 시킨다.  size = 0; // size는 0으로 초기화 시켜준다.  }  SortedArray(int a[], int size){  p = a; // p가 이를 가리키게 하고, a를 해당 배열에 복사한다.  this->size = size; // size 크기의 동적 배열 할당 후,  }  ~SortedArray() { // 소멸자  cout << "소멸함" << endl;  }  void show(); // 배열의 원소를 순서대로 출력  friend SortedArray operator + (SortedArray a, SortedArray b); // SortedArray 객체를 더하는 연산자 => a+b  };  void SortedArray::show() {  cout << "배열 출력 : ";  for (int i = 0; i < size;i++) { // 배열에 들어 있는 값을 비교하여 오름차순으로 정렬  cout << p[i] << " ";  }  cout << endl;  }  SortedArray operator + (SortedArray a, SortedArray b) {  SortedArray tmp;  tmp.size = a.size + b.size; // a 와 b 의 배열 크기를 더해 tmp 배열 크기로 할당  int\* tmp\_array = (int\*)malloc(sizeof(int) \* tmp.size); // a 와 b를 더한 배열 크기를 동적할당  tmp.p = tmp\_array; // 할당 된 배열을 객체 p에 할당    int i = 0, j = 0, index = 0; // 각 배열들의 고유 인덱스  while (i < a.size && j < b.size) { // 배열 a의 크기가 i 보다 크면서 배열 b의 크기가 j보다  클 때까지 while 문 반복  if (a.p[i] < b.p[j]) { // 만일, 배열 a의 값이 배열 b의 값 보다 작을경우  tmp.p[index++] = a.p[i++]; // tmp객체의 p의 index 번째의 a의 p배열의 i번  째 값을 할당  }  else { // 만약 그렇지 않을 경우  tmp.p[index++] = b.p[j++]; // tmp객체의 p의 index 번째의 b의 p배열의 j번  째 값을 할당  }  }  // a 또는 b의 배열의 남은 값들을 모두 tmp의 객체의 배열에 할당  if (i < a.size) {  tmp.p[index++] = a.p[i++];  }  if (j < b.size) {  tmp.p[index++] = b.p[j++];  }  return tmp;  }    int main() {  // 오름차순으로 정렬된 배열  int p[] = { 2,6,20 };  int q[] = { 7,8,10,20,30 };  SortedArray a(p, 3), b(q, 5), c;  c = a + b; // + 연산 수행 결과로, a 와 b에 포함된 배열 정렬을 유지하면서  // 합쳐진 배열을 포함하는 SortedArray 객체를 생성하고, 이를 c에 할당해준다.  a.show(); // a에 포함된 배열 요소를 순서대로 출력  b.show(); // b에 포함된 배열 요소를 순서대로 출력  c.show(); // c에 포함된 배열 요소를 순서대로 출력  return 0;  } |

실행 결과



느낀 점

각 배열에 들어있는 요소를 비교하고 정렬하여 출력해내는 게 신기하기도 했고, 평소에 아무런 생각 없이 편하게 쓰던 정렬 기법을 이렇게 직접 짜보니 감회가 새로웠습니다.