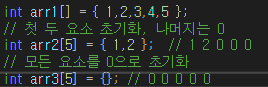
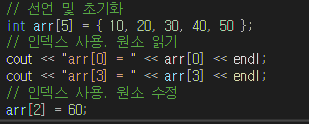
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 33주차 | **기간** | 2025.02.04  ~2025.02.10 | **지도교수** |  |
| 이번주 한일 요약 | 코딩테스트 도서 part5. 배열 | | | | |

프로그래머스 도서 - 둘째마당 : 코딩테스트 완전 정복  
part5. 배열 (133p~175p)

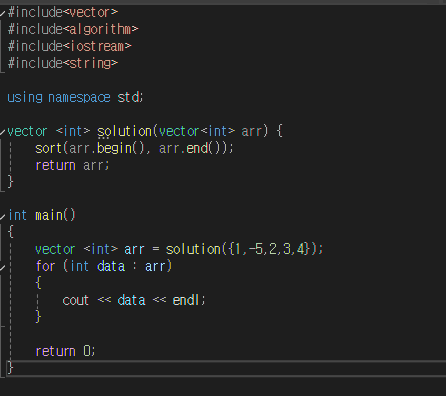
배열 : 하나의 변수 이름으로 동일한 타입의 데이터를 그룹화하여 관리. 인덱스로 데이터에 접근  
  
배열 선언

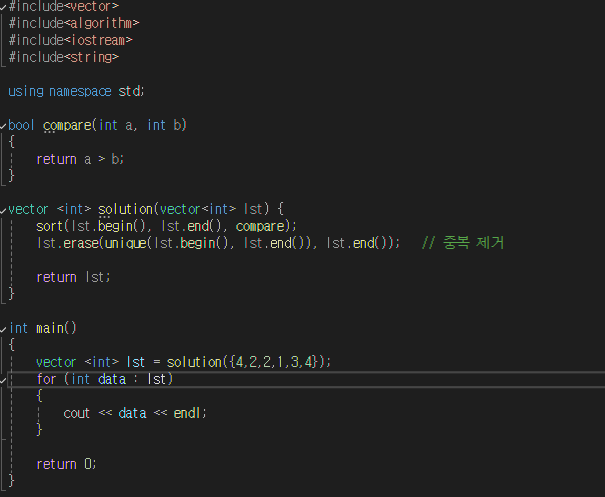


( {}도 없을 경우 쓰레기 값)  
  
배열과 차원  
2차원, 3차원 .... 실제로는 1차원 공간에 저장 => 배열은 차원과 무관하게 메모리에 연속할당.  
- 접근 및 제어 :

  
  
- 2차원 배열 : 행과 열을 명시해 []를 2개연이어 사용. 접근 및 제어는 1차와 동일  


배열의 효율성  
  
- 배열 연산의 시간 복잡도  
배열은 임의접근으로 모든위치의 데이터에 한번에 접근가능 - 접근을 위한 시간복잡도는 O(1)  
  
- 배열에 데이터 추가  
1. 맨 뒤에 삽입할 경우 : 다른 데이터위치에 영향x. 바로 임의접근하여 O(1)  
2. 맨 앞에 삽입할 경우 : 기존 데이터들을 한 칸씩 밀어냄. 데이터개수N, O(N)  
3. 중간에 삽입: 삽입한 데이터 뒤의 개수만큼 미는 연산. 밀어야할 데이터 개수N, O(N)  
  
- 배열을 선택할 때 고려할 점  
1. 할당할 수 있는 메모리 크기 확인. 보통 정수형은 최대 1차원배열은 1,000만개, 2차는 3000\*3000   
2. 중간에 데이터 삽입이 많은지. 배열은 선형자료구조. 중간이나 처음에 삽입 시 시간복잡도가 높아짐.  
but 데이터에 자주 접근하거나 읽어야 하는 부분은 배열이 좋은 성능을 낼 수 있음.  
C++로 코딩테스트에 임할 시 직접 배열 사용보다는 STL의 벡터 사용

몸풀기 문제  
- 정수배열 arr을 오름차순으로 정렬하여 반환하는 solution()함수를 완성하시오.  
제약조건  
- arr길이는 2이상 10^5이하입니다.  
- arr원소값은 -100,000 이상 100,000 이하입니다.  
입출력 예  
arr : [1,-5,2,4,3] / [2,1,1,3,2,5,4] / [6,1,7]  
result : [-5,1,2,3,4]/ [1,1,2,2,3,4,5 / [1,6,7]  
프로토타입  
vector<int> solution(vector<int> arr)  


몸풀기 문제2 : 배열 제어하기  
정수배열 lst. 중복값을 제거하고 배열 데이터를 내림차순으로 정렬해서 반환하는 solution()함수를 구현하세요.  
제약조건  
- lst의 길이는 2이상 100,000 이하입니다.  
- lst의 원소값은 -100,000 이상 100,000 이하입니다.  
프로토 타입 : vector<int>solution(vector<int> lst)  


sort() : 인자로 정렬의 시작, 끝 + 정렬기준  
unique(), erase() 활용하여 중복값 제거

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 34주차 | **다음기간** | 2025.02.11  ~2025.02.17 |
| **다음주 할일** |  | | |
| **지도 교수**  **Comment** |  | | |