C++ 퀴즈 (2023-01-15)

* 조회: 가끔 발생
* 정렬: 항상 이뤄져야 함
* 삽입: 빈번함
* 삭제: 빈번함

1. 다음과 같은 상황에서 vector와 list중 더 적절한 컨테이너를 선택하고 그 이유를 서술하시오.

벡터는 삽입 삭제 시 맨 끝에 넣지 않는 한 O(n)의 시간이 걸리지만 리스트는 삽입 삭제에서 O(1)의 시간이 걸리므로 삽입 삭제가 빈번한 경우 List를 사용하는 것이 적절하다

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 출력 결과와 그 이유는?

10, 20 \n 30, 40

값을 바꾸는 for문에서 &를 취하지 않았기에 복사본의 값만 바꾼 것이기 때문이다

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드는 v 벡터의 원소를 역순으로 출력하는 코드이다. 이 코드의 잘못된 점을 서술하시오.

rbegin은 마지막 원소를, rend는 첫번째 원소의 이전을 가리키고 있고

rbegin, rend 반복자는 자체적으로 ++연산을 --연산으로,

--연산을 ++연산으로 바꾸기 때문에 --a 를 ++a 로 수정해야 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드를 실행해 보고, 다음 코드의 잘못된 점을 서술하시오.

출력하는 for문에서 \*a->begin() 은 \*(a->begin()) 과 같이 처리된다.

위에서 이중 반복문에 들어갔을 때 \*a->begin() 은 첫번째 벡터의 첫번째 값이므로 3

\*a->end()는 첫번째 벡터의 마지막원소 뒤의 값이므로 쓰레기 값이 된다.

따라서 auto b 는 int b 이고 이를 ++하면서 비교하므로 반복자의 비교가 아닌 값의 비교가 되고 값에 ++하면서 출력하는 것이므로 3부터 쓰레기값과 같을때까지 증가하면서 출력하는 문장이 된다

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드를 실행해 보고, 잘못된 점에 대해 생각해 보자.

Sort는 내부적으로 반복자에 –연산을 이용해 중간을 구한다

List의 반복자는 -연산을 할 수 없으므로 알고리즘 헤더에 있는 sort를 사용할 수 없다.

텍스트이(가) 표시된 사진

자동 생성된 설명

5 – 번외. 5번 컨테이너와의 차이에 대해 생각해 보자.

덱은 리스트와 다르게 랜덤 액세스가 가능한 컨테이너 이므로 반복자에 -연산을 해서 중간을 구할 수 있다. 따라서 알고리즘 헤더에 있는 sort를 사용할 수 있다