**Homework 8 컴퓨터공학부 202211390 최준원**

|  |
| --- |
| Q1 |
| Source Code |
| #include <iostream>  using namespace std;  int intarray[5] = { 3, 7, 2, 12, 14 };  double doublearray[5] = { 22.7, 14.2, 3.8, 12.23, 11.2 };  char chararray[6] = { 'C', 'a', 'B', 'E', 'N', 'Q' };  string stringarray[4] = { "John", "Lu", "Mary", "Su" };  template <typename T, int N>  void reverse(T(&array)[N]) {  for (int i = 0; i < N / 2; i++) {  swap(array[i], array[N - i - 1]);  }  }  template <typename T>  void swap(T& first, T& second) {  T temp = first;  first = second;  second = temp;  }  template <typename T, int N>  void print(T(&array)[N])  {  for (int i = 0; i < N; i++)  {  cout << array[i] << " ";  }  cout << endl;  }  int main() {  cout << "Original Array" << endl;  print(intarray);  reverse(intarray);  cout << "Reversed Array" << endl;  print(intarray);  cout << endl << endl;  cout << "Original Array" << endl;  print(doublearray);  reverse(doublearray);  cout << "Reversed Array" << endl;  print(doublearray);  cout << endl << endl;  cout << "Original Array" << endl;  print(chararray);  reverse(chararray);  cout << "Reversed Array" << endl;  print(chararray);  cout << endl << endl;  cout << "Original Array" << endl;  print(stringarray);  reverse(stringarray);  cout << "Reversed Array" << endl;  print(stringarray);  cout << endl << endl << endl;  cout << "#-- Custom Test Cases --" << endl;    cout << endl << "//bool type array reverse" << endl;  bool boolarray[5] = { true, false, false, true, true };  cout << "Original Array" << endl;  print(boolarray);  reverse(boolarray);  cout << "Reversed Array" << endl;  print(boolarray);  cout << endl;    cout << endl << "//short type array reverse" << endl;  short shortarray[9] = { -21, -23, -32768, -1, 35, 12124, 2, 325, 32768};  cout << "Original Array" << endl;  print(shortarray);  reverse(shortarray);  cout << "Reversed Array" << endl;  print(shortarray);  cout << endl << endl;  return 0;  } |
| Screenshot |
| 텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진  자동 생성된 설명 |

|  |
| --- |
| Q2 |
| Code |
| #include <iostream>  #include <string>  #include <exception>  using namespace std;  template <typename T>  class Array {  private:  T\* Arr;  int size, count = 0;  public:  Array(int n);  ~Array();  void add(T data);  void print();  };  template <typename charT>  class basic\_string {  };  class Except : public out\_of\_range {  public:  Except(const string& message);  };  Except::Except(const string& message)  :out\_of\_range(message)  {  }  template<typename T>  Array<T>::Array(int n)  :size(n)  {  Arr = new T[size];  }  template<typename T>  Array<T>::~Array()  {  delete[] Arr;  Arr = 0;  }  template<typename T>  void Array<T>::add(T data)  {  Arr[count] = data;  count++;  if (count == size) {  Except exception("Array is full.\n");  throw exception;  }  }  template<typename T>  void Array<T>::print()  {  for (int i = 0; i < count; i++) {  cout << Arr[i] << " ";  }  cout << endl;  }  int main()  {  cout << "Instantiation of an array of integers." << endl;  Array <int> array1(5); // set the size of the array using ‘new’  try  {  array1.add(-5);  array1.add(7);  array1.add(8);  array1.add(10);  array1.add(14);  array1.add(20);  array1.add(-14);  }  catch (out\_of\_range& ex)  {  cout << ex.what();  }  array1.print();  cout << "Instantiation of an array of doubles." << endl;  Array <double> array2(5); // set the size of the array  try  {  array2.add(5.3);  array2.add(7.6);  array2.add(8.1);  array2.add(-1);  }  catch (out\_of\_range& ex)  {  cout << ex.what();  }  array2.print();    cout << "Instantiation of an array of strings." << endl;  Array <string> array3(4); // set the size of the array  try  {  array3.add("John");  array3.add("Lu");  array3.add("Mary");  array3.add("Leo");  array3.add("Robert");  }  catch (out\_of\_range& ex)  {  cout << ex.what();  }  array3.print();    cout << "\n#-- Custom Test Cases -- \n\n";  cout << "Instantiation of an array of characters." << endl;  Array <char> array4(9); // set the size of the array  try  {  array4.add('a');  array4.add('G');  array4.add('D');  array4.add('e');  array4.add('K');  }  catch (out\_of\_range& ex)  {  cout << ex.what();  }  array4.print();  cout << endl;  cout << "Instantiation of an array of booleans." << endl;  Array <bool> array5(7); // set the size of the array  try  {  array5.add(true);  array5.add(false);  array5.add(false);  array5.add(true);  array5.add(true);  array5.add(false);  array5.add(true);  array5.add(false);  }  catch (out\_of\_range& ex)  {  cout << ex.what();  }  array5.print();  array1.~Array();  array2.~Array();  array3.~Array();  array4.~Array();  array5.~Array();  return 0;  } |
| Screenshot |
| 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명 |