**202211341 컴퓨터공학부 이윤희**

|  |
| --- |
| **Question1** |
| **Source code** |
| #include <iostream>  #include <fstream>  #include <vector>  bool SearchFromFile(const std::string& file\_name, int target)  {  std::ifstream file;  file.open(file\_name);  if (!file.is\_open())  {  std::cerr << "Failed to open the file." << std::endl;  return false;  }  int num;  while (file >> num)  {  if (num == target)  {  file.close();  return true;  }  }  file.close();  return false;  }  int main()  {  std::vector<int> nums = { 17, 72, 73 };  for (auto i = 0; i < nums.size(); i++)  {  int value = nums[i];  std::cout << "Search the integer to look for: " << value << std::endl;  if (SearchFromFile("File1.txt", value))  {  std::cout << value << " is in the file." << std::endl;  }  else  {  std::cout << value << " is not in the file!" << std::endl;  }  }  std::cout << "\nPress Any Key to Terminate ...\n";  getchar();  return 0;  } |
| **Output** |
| **Question2** |
| **Source code** |
| #include <iostream>  #include <string>  template <typename T>  class List {  private:  struct Node {  T data;  Node\* next;  };  Node\* head;  public:  List() : head(nullptr) {}  void insert(int index, const T& value) {  Node\* newNode = new Node;  newNode->data = value;  if (index == 0 || head == nullptr) {  newNode->next = head;  head = newNode;  }  else {  Node\* current = head;  for (int i = 0; i < index - 1; i++) {  if (current->next == nullptr) {  break;  }  current = current->next;  }  newNode->next = current->next;  current->next = newNode;  }  }  void print() const {  Node\* current = head;  while (current != nullptr) {  std::cout << current->data << "\n";  current = current->next;  }  }  void reversePrint() const {  if (head == nullptr) {  return;  }  Node\* current = head;  Node\* prev = nullptr;  Node\* next = nullptr;  while (current != nullptr) {  next = current->next;  current->next = prev;  prev = current;  current = next;  }  current = prev;  while (current != nullptr) {  std::cout << current->data << "\n";  current = current->next;  }  }  };  int main() {  List<int> list1;  list1.insert(0, 17);  list1.insert(1, 23);  list1.insert(2, 11);  list1.insert(3, 18);  std::cout << "Printing the list1" << std::endl;  list1.print();  std::cout << std::endl << std::endl;  std::cout << "Printing the list1 in reverse order" << std::endl;  list1.reversePrint();  std::cout << std::endl << std::endl;  List<std::string> list2;  list2.insert(0, "Michael");  list2.insert(1, "Jane");  list2.insert(2, "Sophie");  list2.insert(3, "Thomas");  list2.insert(4, "Rose");  list2.insert(5, "Richard");  std::cout << "Printing the list2" << std::endl;  list2.print();  std::cout << std::endl << std::endl;  std::cout << "Printing the list2 in reverse order" << std::endl;  list2.reversePrint();  std::cout << std::endl << std::endl;  std::cout << "\nPress Any Key to Terminate ...\n";  getchar();  return 0;  } |
| **Output** |
| **Question3** |
| **Source code**  #include <string>  #include <iostream>  #include <vector>  #include <algorithm>  int main() {  std::vector<int> integers;  std::vector<std::string> strings;  strings.push\_back("John");  strings.push\_back("Mary");  strings.push\_back("Lucie");  strings.push\_back("Robert");  strings.push\_back("Suzan");  strings.push\_back("Richard");  integers.push\_back(17);  integers.push\_back(23);  integers.push\_back(11);  integers.push\_back(18);  std::cout << "Unsorted Names: " << std::endl;  std::for\_each(strings.begin(), strings.end(), [](std::string value) {  std::cout << value << " ";  });  std::cout << std::endl << std::endl;  std::sort(strings.begin(), strings.end());  std::cout << "Sorted Names: "<< std::endl;  std::for\_each(strings.begin(), strings.end(), [](std::string value) {  std::cout << value << " ";  });  std::cout << std::endl<< std::endl;  std::cout << "Unsorted Integers: "<< std::endl;  std::for\_each(integers.begin(), integers.end(), [](int value) {  std::cout << value << " ";  });  std::cout << std::endl<< std::endl;  std::sort(integers.begin(), integers.end());  std::cout << "Sorted Integers: "<< std::endl;  std::for\_each(integers.begin(), integers.end(), [](int value) {  std::cout << value << " ";  });  std::cout << std::endl;  return 0;  } |
| **Output** |