Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



# Звіт

про виконання

**Лабораторних та практичних робіт №5*:*** «Мови та парадигми програмування»

***з розділу***: «Лінійні та розгалужені алгоритми. Умовні оператори. Константи,

змінні»

***Виконав:***

студент групи ШІ-11 Сабадило Марко

Львів 2023

1. Файли С++:

У С++ існують засоби для роботи з файлами, такі як введення/виведення, читання та запис. Це дозволяє програмам взаємодіяти з зовнішніми файлами, зберігаючи та зчитуючи дані.

2. Системи числення:

Системи числення включають десяткову (основа 10), двійкову (основа 2), вісімкову (основа 8) та шістнадцяткову (основа 16). Кожна система числення використовує певні символи для представлення чисел у відповідній основі.

3. Бінарні файли:

Бінарні файли зберігають дані у вигляді бінарних (або двійкових) форматів, що дозволяє ефективно зберігати та читати структуровані дані, такі як числа, об'єкти тощо.

4. Символи і рядкові змінні:

Символи (char) та рядки (std::string) в С++ використовуються для представлення текстової інформації. Символи можна використовувати для представлення одного символу, тоді як рядки дозволяють зберігати послідовності символів.

5. Текстові файли:

Текстові файли є файлами, в яких дані зберігаються у вигляді тексту. Стандартна бібліотека С++ містить засоби для читання та запису текстових файлів, що дозволяє обробляти структурований або неструктурований текст.

6. Стандартна бібліотека та робота з файлами:

Стандартна бібліотека С++ надає різноманітні функції для роботи з файлами, такі як fstream для текстових файлів та ifstream/ofstream для бінарних файлів. Ці інструменти дозволяють читати, записувати, переміщати та інші операції з файлами.

7. Створення та використання бібліотек:

В С++ можна створювати власні бібліотеки, які включають в себе набір класів, функцій та інших компонентів. Це дозволяє реюзати код між різними проектами, забезпечуючи модульність та організацію програмного

**Код програм:**

Завдання № 1

|  |
| --- |
| #include <iostream> #include <vector> using namespace std; vector<int> bubble\_sort(vector<int> arr,int n){  for(int i=0;i<n-1;i++){  for(int j=0;j<n-1-i;j++){  if(arr[j]>arr[j+1]) swap(arr[j],arr[j+1]);  }  }  return arr; }  vector<int> removeDuplicates(vector<int> arr) {  for (int i=1 ;i <arr.size()-1;i++){  if (arr[i]==arr[i+1]||arr[i]==arr[i-1]){  arr.erase(arr.begin()+i);  i--;  }  }  return arr; } vector<int> rotateArray(vector<int>arr, int k) {  for(int i=0; i<k; i++){  for(int j=0; j<arr.size()-1; j++){  swap(arr[j],arr[j+1]);  }  }  return arr; } int main() {  int n,k;   // Введення розміру масиву та значення k  cin >> n >>k;  if (n<1 ||n>1000||k<1 ||k>1000 )return 0;   vector<int> inputArray;  // Введення елементів масиву  for (int i = 0; i < n; ++i) {  int a;  cin >> a;  if (a < 0 || a > 100)return 0;  inputArray.push\_back(a);  }  inputArray = bubble\_sort(inputArray,n);  inputArray = removeDuplicates(inputArray);  inputArray = rotateArray(inputArray,k);   cout << inputArray.size() << endl;  for (int i = 0; i <inputArray.size(); ++i) {  cout << inputArray[i] << " ";   }  cout << endl;   return 0; } |

Завдання № 2

|  |
| --- |
| #include <iostream> #include <vector> #include <algorithm>  using namespace std;  int main() {  // Зчитування вхідних даних  int N, K;  cin >> N >> K;   vector<int> a(N);  for (int i = 0; i < N; ++i) {  cin >> a[i];  }   // Видалення повторюючихся елементів  sort(a.begin(), a.end());  auto last = unique(a.begin(), a.end());  a.erase(last, a.end());   // Обертання та виведення результату  rotate(a.begin(), a.begin() + K % a.size(), a.end());   // Виведення результату  cout << a.size() << endl;  for (int i = 0; i < a.size(); ++i) {  cout << a[i];  if (i < a.size() - 1) {  cout << " ";  }  }   return 0; } |

Завдання № 3

|  |
| --- |
| #include <stdio.h> #include <string.h> #include <iostream> using namespace std; void remove\_even\_words(char \*s) {  char result[256] = ""; // для зберігання зміненого рядка  char \*token; // для розбиття рядка на слова  int count = 0; // лічильник для слів   // Розбиваємо рядок на слова  token = strtok(s, " ");  while (token != NULL) {  count++;   // Якщо номер слова парний, пропускаємо його  if (count % 2 != 0) {  strcat(result, token);  strcat(result, " "); // додаємо пробіл після кожного слова  }   token = strtok(NULL, " "); // отримуємо наступне слово  }   // Видаляємо останній пробіл і додаємо крапку  result[strlen(result) - 1] = '.';  strcpy(s, result); }  int main() {  char s[256];  gets(s);   remove\_even\_words(s);   cout<< s;   return 0; } |

Завдання № 3.2

|  |
| --- |
| #include <iostream> #include <fstream> #include <string> #include <vector>  using namespace std;  struct Car {  string brand;  int serialNumber;  string regNumber;  int year; };  void manipulateFile() {  fstream file("cars.bin", ios::binary | ios::in | ios::out | ios::trunc);   if (!file.is\_open()) {  cerr << "Error opening file" << endl;  return;  }   vector<Car> cars = **{** {"BMW", 12345, "AA1234BB", 2010},  {"Toyota", 67890, "CC5678DD", 2015},  {"Ford", 44556, "GG1122HH", 2020},  {"Mercedes", 77889, "II3344JJ", 2019}  **}**;   file.write(reinterpret\_cast<const char\*>(cars.data()), cars.size() \* sizeof(Car));   file.seekg(0, ios::beg);  Car tempCar;  cout << "File content:\n";  while (file.read(reinterpret\_cast<char\*>(&tempCar), sizeof(Car))) {  cout << "Brand: " << tempCar.brand << ", Registration Number: " << tempCar.regNumber << endl;  }   file.seekp(3 \* sizeof(Car), ios::beg);  int newCount = 2;  while (file.read(reinterpret\_cast<char\*>(&tempCar), sizeof(Car))) {  file.seekp(-sizeof(Car), ios::cur);  file.write(reinterpret\_cast<const char\*>(&tempCar), sizeof(Car));  file.seekp(sizeof(Car), ios::cur);  }  file.close();   fstream("cars.bin", ios::binary | ios::in | ios::out).close();  file.open("cars.bin", ios::binary | ios::in | ios::out);  file.seekp(0, ios::end);  file.write(reinterpret\_cast<const char\*>(&tempCar), sizeof(Car));   file.close(); }   int main() {  manipulateFile();  return 0; } |
|  |

Завдання № 4

|  |
| --- |
| #include <iostream> #include <fstream> #include <string> #include <sstream> #include <vector> #include <cctype> #include <random>  using namespace std;  // Функція для підрахунку голосних букв у слові int countVowels(const string& word) {  int count = 0;  for (char c : word) {  if (tolower(c) == 'a' || tolower(c) == 'e' || tolower(c) == 'i' || tolower(c) == 'o' || tolower(c) == 'u') {  count++;  }  }  return count; }  // Функція для генерації випадкових слів string generateRandomWord(int length) {  static const char alphabet[] = "abcdefghijklmnopqrstuvwxyz";  string word;  for (int i = 0; i < length; ++i) {  word += alphabet[rand() % (sizeof(alphabet) - 1)];  }  return word; }  int main() {  // Створення файлу F1 та запис 10 рядків у нього  ofstream fileF1("F1.txt");  if (!fileF1.is\_open()) {  cerr << "Unable to open F1.txt\n";  return 1;  }   srand(time(nullptr)); // Ініціалізація генератора випадкових чисел   for (int i = 0; i < 10; ++i) {  fileF1 << generateRandomWord(5) << " " << generateRandomWord(4) << " " << generateRandomWord(6) << endl;  }  fileF1.close();   // Створення файлу F2 для запису рядків  ofstream fileF2("F2.txt");  ifstream readFile("F1.txt");   if (!fileF2.is\_open() || !readFile.is\_open()) {  cerr << "Unable to open one of the files.\n";  return 1;  }   string line;  int wordNumberWithMaxVowels = -1;  int maxVowelCount = 0;   while (getline(readFile, line)) {  istringstream iss(line);  string word;  vector<string> words;   while (iss >> word) {  words.push\_back(word);  }   if (words.size() > 2) {  fileF2 << line << endl;   for (size\_t i = 0; i < words.size(); ++i) {  int vowelCount = countVowels(words[i]);  if (vowelCount > maxVowelCount) {  maxVowelCount = vowelCount;  wordNumberWithMaxVowels = i + 1; // Номер слова починається з 1  }  }  }  }   readFile.close();  fileF2.close();   if (wordNumberWithMaxVowels != -1) {  cout << "The word number with the most vowels is: " << wordNumberWithMaxVowels << endl;  } else {  cout << "No words with vowels found.\n";  }   return 0; } |

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис