

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.
Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання
бібліотек.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 6
ВНС Лабораторної Роботи № 8
ВНС Лабораторної Роботи № 9
Алготестер Лабораторної Роботи №4
Алготестер Лабораторної Роботи №6
Практичних Робіт до блоку №5

Виконав:
Студент групи ШІ-12
Роман Бісюк

Львів 2024

Тема роботи: Файлова система в C++. Робота з бінарними файлами та текстовими файлами, маніпуляції символами й рядковими змінними, як типу `std::string`, так і `char*`. Ознайомлення з можливостями стандартної бібліотеки C++ для роботи з файлами та створенням власних бібліотек для розширення функціональності.

Мета роботи: Опанувати практичні навички роботи з файлами в мові C++: створення, зчитування та запис даних у бінарні й текстові файли. Засвоїти принципи роботи з рядковими змінними різних типів (`std::string` і `char*`), вивчити використання стандартних методів та функцій для маніпуляцій з ними. Дослідити основи створення та застосування власних бібліотек для зручності повторного використання коду й розширення можливостей стандартної бібліотеки C++.

Джерела:

CS50 course

University lectures

Google + chatGPT: string functions and memory allocation

Виконання роботи:


```

1  #include <iostream>
2  #include <string>
3  #include <climits>
4  using namespace std;
5
6  int main()
7  {
8      cout << "Введіть рядок для знаходження найменшого слова:\n";
9      string s;
10     getline(cin, s);
11
12     int counter = 0, min_length = INT_MAX;
13
14     for (char c : s)
15     {
16         if (c != ' ')
17             counter++;
18         else
19         {
20             if (counter > 0)
21                 min_length = min(min_length, counter);
22             counter = 0;
23         }
24     }
25
26     if (counter > 0)
27         min_length = min(min_length, counter);
28
29     if (min_length == INT_MAX)
30         min_length = 0;
31
32     cout << min_length;
33
34     return 0;
35 }

```

Введіть рядок для знаходження найменшого слова:
Купити мішок горіхів для нашого бенкету
3

Task 4:

Lab# programming: VNS Lab 8

time expected – 2 hours

time spent – 4 hours

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <cstring>
4 using namespace std;
5
6 struct Stadion {
7     char name[40];
8     int year;
9     int platforms;
10    char sport[40];
11};
12
13 void input_stadion(Stadion &st) {
14    cout << "Enter name: ";
15    cin >> st.name;
16    cout << "Enter year: ";
17    cin >> st.year;
18    cout << "Enter number of platforms: ";
19    cin >> st.platforms;
20    cout << "Enter type of sport: ";
21    cin >> st.sport;
22}
23
24 void print_stadion(const Stadion &st) {
25    cout << "Name: " << st.name << endl;
26    cout << "Year: " << st.year << endl;
27    cout << "Number of platforms: " << st.platforms << endl;
28    cout << "Sport: " << st.sport << endl;
29}
30
31 void add_stadion(const char *filename) {
32    FILE *file = fopen(filename, "ab");
33    if (!file) {
34        cerr << "Error opening file for appending!" << endl;
35        return;
36    }
37    Stadion st;
38    input_stadion(st);
39    fwrite(&st, sizeof(Stadion), 1, file);
40    fclose(file);
41}
42
43 void show_all(const char *filename) {
44    FILE *file = fopen(filename, "rb");
45    if (!file) {
46        cerr << "Error opening file for reading!" << endl;
47        return;
48    }
49    cout << "List of stadiums:\n";
50    Stadion st;
51    while (fread(&st, sizeof(Stadion), 1, file)) {
52        print_stadion(st);
53        cout << "-----\n";
54    }
55    fclose(file);
56}
57
58 void delete_old_stadions(const char *filename, int target_year) {
59    FILE *file = fopen(filename, "rb");
60    FILE *temp = fopen("temp.dat", "wb");
61    if (!file || !temp) {
62        cerr << "Error opening file for deletion!" << endl;
63        return;
64    }
65    Stadion st;
66    while (fread(&st, sizeof(Stadion), 1, file)) {
67        if (st.year >= target_year) {
68            fwrite(&st, sizeof(Stadion), 1, temp);
69        }
70    }
71    fclose(file);
72    fclose(temp);
73    remove(filename);
74    rename("temp.dat", filename);
75    cout << "Old stadiums deleted.\n";
76}
77
78 void add_before_position(const char *filename, int position, const Stadion &st1, const Stadion &st2) {
79    FILE *file = fopen(filename, "rb");
80    FILE *temp = fopen("temp.dat", "wb");
81    if (!file || !temp) {
82        cerr << "Error opening file for updating!" << endl;
83        return;
84    }
85    Stadion st;
86    int current = 1;
87    while (fread(&st, sizeof(Stadion), 1, file)) {
88        if (current == position) {
89            fwrite(&st1, sizeof(Stadion), 1, temp);
90            fwrite(&st2, sizeof(Stadion), 1, temp);
91            fwrite(&st, sizeof(Stadion), 1, temp);
92            current++;
93        }
94    }
95    fclose(file);
96    fclose(temp);
97    remove(filename);
98    rename("temp.dat", filename);
99    cout << "New stadiums added.\n";
100}
101
102 int main() {
103    const char *filename = "Database.dat";
104    FILE *file = fopen(filename, "wb");
105    if (!file) {
106        cerr << "Error creating file!" << endl;
107        return 1;
108    }
109    fclose(file);
110
111    int stadion_count;
112    cout << "Enter the number of stadiums you want to create: ";
113    cin >> stadion_count;
114
115    for (int i = 0; i < stadion_count; i++) {
116        cout << "Enter details for stadion " << i + 1 << ":\n";
117        add_stadion(filename);
118    }
119
120    int year;
121    cout << "Enter the minimum year to keep: ";
122    cin >> year;
123    delete_old_stadions(filename, year);
124
125    Stadion st1, st2;
126    int position;
127    cout << "Enter position to insert before: ";
128    cin >> position;
129    cout << "Enter 2 new stadiums:\n";
130    input_stadion(st1);
131    input_stadion(st2);
132    add_before_position(filename, position, st1, st2);
133
134    show_all(filename);
135
136    return 0;
137}
```

```
Enter the number of stadions you want to create: 2
Enter details for stadion 1:
Enter name: OLYMP
Enter year: 1997
Enter number of platforms: 2
Enter type of sport: Football
Enter details for stadion 2:
Enter name: Liverpool
Enter year: 2001
Enter number of platforms: 1
Enter type of sport: valleyball
Enter the minimum year to keep: 2000
Old stadions deleted.
Enter position to insert before: 2
Enter 2 new stadions:
Enter name: Roman
Enter year: 1111
Enter number of platforms: 2
Enter type of sport: regby
Enter name: firststone
Enter year: 1231
Enter number of platforms: 4
Enter type of sport: hockey
New stadions added.
```

```
List of stadions:
Name: Liverpool
Year: 2001
Number of platforms: 1
Sport: valleyball
-----
```

Task5:

Lab# programming: VNS Lab 9

time expected - 2 hours

time spent – 2 hours

```
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <string>
5 #include <vector>
6 #include <cctype>
7 #include <algorithm>
8
9 using namespace std;
10
11 bool has_duplicate_words(const string &line) {
12     istringstream iss(line);
13     vector<string> words;
14     string word;
15
16     while (iss >> word) {
17         if (find(words.begin(), words.end(), word) != words.end()) {
18             return true;
19         }
20         words.push_back(word);
21     }
22     return false;
23 }
24
25 void copy_lines_with_duplicates(ifstream &F1, ofstream &F2) {
26     string line;
27     while (getline(F1, line)) {
28         if (has_duplicate_words(line)) {
29             F2 << line << endl;
30         }
31     }
32 }
33
34 void find_word_with_max_a(ifstream &F1) {
35     F1.clear();
36     F1.seekg(0);
37     string line;
38     int line_number = 1;
39
40     while (getline(F1, line)) {
41         istringstream iss(line);
42         string word;
43         int max_a_count = 0, max_word_index = -1, word_index = 0;
44
45         while (iss >> word) {
46             int a_count = count_if(word.begin(), word.end(), [](char ch) {
47                 return tolower(ch) == 'a';
48             });
49
50             if (a_count > max_a_count) {
51                 max_a_count = a_count;
52                 max_word_index = word_index + 1;
53             }
54             ++word_index;
55         }
56
57         if (max_word_index > 0) {
58             cout << "line " << line_number << ": word #" << max_word_index
59                 << " has the most 'A' (" << max_a_count << ")." << endl;
60         }
61         ++line_number;
62     }
63 }
64
65 int main() {
66     string file1 = "F1.txt";
67     string file2 = "F2.txt";
68
69     ofstream F1(file1);
70     if (!F1.is_open()) {
71         cerr << "Error creating file F1" << endl;
72         return 1;
73     }
74
75     cout << "Enter at least 10 lines for file F1:" << endl;
76     string line;
77     for (int i = 0; i < 10; ++i) {
78         getline(cin, line);
79         F1 << line << endl;
80     }
81     F1.close();
82
83     ifstream F1_read(file1);
84     ofstream F2(file2);
85     if (!F1_read.is_open() || !F2.is_open()) {
86         cerr << "Error opening files" << endl;
87         return 1;
88     }
89
90     copy_lines_with_duplicates(F1_read, F2);
91
92     F1_read.clear();
93     F1_read.seekg(0);
94     find_word_with_max_a(F1_read);
95
96     F1_read.close();
97     F2.close();
98
99     cout << "Tasks completed. Check F2.txt for results." << endl;
100    return 0;
101 }
```

```
Enter at least 10 lines for file F1:
Text messaging, or simply texting, is the act of composing
and sending electronic messages, typically consisting
of alphabetic and numeric characters, between two or more users
of mobile phones, tablet computers
smartwatches, desktops/laptops, or another type of compatible computer
Text messages may be sent over a cellular network or may also be sent via satellite or Internet connection.
The term originally referred to messages sent using the Short Message Service
(SMS) on mobile devices. It has grown beyond alphanumeric text
to include multimedia messages using the Multimedia Messaging Service (MMS)
and Rich Communication Services (RCS), which can contain digital images
Line 1: word #2 has the most 'A' (1).
Line 2: word #1 has the most 'A' (1).
Line 3: word #2 has the most 'A' (2).
Line 4: word #4 has the most 'A' (1).
Line 5: word #1 has the most 'A' (2).
Line 6: word #2 has the most 'A' (1).
Line 7: word #3 has the most 'A' (1).
Line 8: word #9 has the most 'A' (2).
Line 9: word #3 has the most 'A' (1).
Line 10: word #1 has the most 'A' (1).
Tasks completed. Check F2.txt for results.
```

```
≡ F1.txt
1 Text messaging, or simply texting, is the act of composing
2 and sending electronic messages, typically consisting
3 of alphabetic and numeric characters, between two or more users
4 of mobile phones, tablet computers
5 smartwatches, desktops/laptops, or another type of compatible computer
6 Text messages may be sent over a cellular network or may also be sent via satellite or Internet connection.
7 The term originally referred to messages sent using the Short Message Service
8 (SMS) on mobile devices. It has grown beyond alphanumeric text
9 to include multimedia messages using the Multimedia Messaging Service (MMS)
10 and Rich Communication Services (RCS), which can contain digital images
11

≡ F2.txt
1 Text messages may be sent over a cellular network or may also be sent via satellite or Internet connection.
2
```

TASK 6:

Lab# programming: Algotester Lab 4

Time expected – 1 hour
time spent - 2 hours

Lab 4v3

Limits: 2 sec., 256 MiB

Вам дано масив, який складається з N додатніх цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2)

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видаліть усі дублікати з масиву.

Виведіть результуючий масив.

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.207	View
a few seconds ago	C++ 23	Accepted	0.003	1.203	View

З бібліотеками STL

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int N;
9      cin >> N;
10
11     vector<int> array(N);
12
13     for (int i = 0; i < N; i++) {
14         cin >> array[i];
15     }
16
17     auto zero = partition(array.begin(), array.end(), [](int i) { return i % 3 == 0; });
18     auto one = partition(zero, array.end(), [](int i) { return i % 3 == 1; });
19
20     sort(array.begin(), zero);
21     sort(zero, one, greater<int>());
22     sort(one, array.end());
23
24     auto end = unique(array.begin(), array.end());
25     array.erase(end, array.end());
26
27     cout << array.size() << endl;
28     for (int x : array) {
29         cout << x << " ";
30     }
31     cout << endl;
32
33     return 0;
34 }
```


Без бібліотек STL

```
1  #include <iostream>
2  using namespace std;
3
4  void sortAsc(int arr[], int start, int end) {
5      for (int i = start; i < end - 1; i++) {
6          for (int j = i + 1; j < end; j++) {
7              if (arr[i] > arr[j]) {
8                  swap(arr[i], arr[j]);
9              }
10         }
11     }
12 }
13
14 void sortDesc(int arr[], int start, int end) {
15     for (int i = start; i < end - 1; i++) {
16         for (int j = i + 1; j < end; j++) {
17             if (arr[i] < arr[j]) {
18                 swap(arr[i], arr[j]);
19             }
20         }
21     }
22 }
23
24 int removeDuplicates(int arr[], int n) {
25     int newSize = 0;
26     for (int i = 0; i < n; i++) {
27         if (i == 0 || arr[i] != arr[newSize - 1]) {
28             arr[newSize++] = arr[i];
29         }
30     }
31     return newSize;
32 }
33
34 int main() {
35     int N;
36     cin >> N;
37
38     int* array = new int[N];
39     for (int i = 0; i < N; i++) {
40         cin >> array[i];
41     }
42
43     int* mod0 = new int[N];
44     int* mod1 = new int[N];
45     int* mod2 = new int[N];
46
47     int count0 = 0, count1 = 0, count2 = 0;
48
49     for (int i = 0; i < N; i++) {
50         if (array[i] % 3 == 0) {
51             mod0[count0++] = array[i];
52         } else if (array[i] % 3 == 1) {
53             mod1[count1++] = array[i];
54         } else {
55             mod2[count2++] = array[i];
56         }
57     }
58
59     sortAsc(mod0, 0, count0);
60     sortDesc(mod1, 0, count1);
61     sortAsc(mod2, 0, count2);
62
63     int newSize = 0;
64     for (int i = 0; i < count0; i++) array[newSize++] = mod0[i];
65     for (int i = 0; i < count1; i++) array[newSize++] = mod1[i];
66     for (int i = 0; i < count2; i++) array[newSize++] = mod2[i];
67
68     newSize = removeDuplicates(array, newSize);
69
70
71     cout << newSize << endl;
72     for (int i = 0; i < newSize; i++) {
73         cout << array[i] << " ";
74     }
75     cout << endl;
76
77     delete[] array;
78     delete[] mod0;
79     delete[] mod1;
80     delete[] mod2;
81
82     return 0;
83 }
```

Task 7:

Lab# programming: Algotester Lab 6

Time expected – 3 hours

time spent – 2.5 hours

Lab 6v2

Limits: 2 sec., 256 MiB

У вас є шахова дошка розміром 8×8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка O
- Пішак P
- Тура R
- Кінь N
- Слон B
- Король K
- Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть Q запитів з координатами клітинки $\{x, y\}$. На кожен запит ви маєте вивести стрічку s_i - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X .

У випадку, якщо клітинку не атакують - виведіть O .

Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.238	View

Task 8 :

Practice# programming: Class Practice Task

time expected – 2 hours

time spent – 2 hours

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <cmath>
5
6  using namespace std;
7
8  bool attacks(char figure, int fx, int fy, int tx, int ty) {
9      switch (figure) {
10         case 'P': return fx == tx - 1 && abs(fy - ty) == 1;           // Пішак
11         case 'R': return fx == tx || fy == ty;                       // Тура
12         case 'N': return (abs(fx - tx) == 2 && abs(fy - ty) == 1) ||
13             (abs(fx - tx) == 1 && abs(fy - ty) == 2);               // Кінь
14         case 'B': return abs(fx - tx) == abs(fy - ty);               // Слон
15         case 'K': return abs(fx - tx) <= 1 && abs(fy - ty) <= 1;     // Король
16         case 'Q': return fx == tx || fy == ty ||
17             abs(fx - tx) == abs(fy - ty);                           // Ферзь
18         default: return false;
19     }
20 }
21
22 int main() {
23     vector<pair<char, pair<int, int>>> figures;
24
25     for (int i = 1; i <= 8; ++i) {
26         string row;
27         cin >> row;
28         for (int j = 0; j < 8; ++j) {
29             if (row[j] != '0') {
30                 figures.push_back({row[j], {i, j + 1}});
31             }
32         }
33     }
34
35     int Q;
36     cin >> Q;
37
38     while (Q-- > 0) {
39         int x, y;
40         cin >> x >> y;
41
42         set<char> attackers;
43         bool occupied = false;
44
45         for (auto &[figure, coords] : figures) {
46             auto &[fx, fy] = coords;
47             if (fx == x && fy == y) {
48                 occupied = true;
49                 break;
50             }
51             if (attacks(figure, fx, fy, x, y)) {
52                 attackers.insert(figure);
53             }
54         }
55
56         if (occupied) {
57             cout << 'X' << endl;
58         } else if (attackers.empty()) {
59             cout << '0' << endl;
60         } else {
61             for (char c : attackers) {
62                 cout << c;
63             }
64             cout << endl;
65         }
66     }
67
68     return 0;
69 }

```

Enter content to write to file: hello world
Success. File created.
Success. File is copied.

≡ file1.txt

≡ file2.txt

≡ file1.txt

1 hello world

Task 9:

Epic 5 Task 9 - Practice# programming: Self Practice Task

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      int k;
8      string s;
9
10     cin >> k;
11     cin >> s;
12
13     string target = "TOILET";
14     int count = 0;
15
16     size_t pos = 0;
17     while ((pos = s.find(target, pos)) != string::npos) {
18         count++;
19         pos += target.size();
20     }
21
22     if (count >= k) {
23         cout << "YES" << endl;
24     } else {
25         cout << "NO" << endl;
26     }
27
28     return 0;
29 }
```

Допоможе чи заб'є?

Обмеження: 2 сек., 256 МБ

До Тойлет-мена, відомого вам білоруського супергероя, дуже часто звертаються по допомогу різні люди. Проте, очевидно, усім мужній гігант допомогти не зможе — не вистачить часу. Та й не дуже хоче.

Саме тому, коли до нього приходить певне SMS-повідомлення з проханням про допомогу, він погодиться допомогти тоді й лише тоді, коли в цьому повідомленні знайдеться хоча б *k* входжень рядка **TOILET**, які не перетинаються.

За заданим повідомленням *s*, яке складається з великих латинських символів, виведіть **YES**, якщо Тойлет-мен погодиться допомогти людині, яка написала це повідомлення. У протилежному разі виведіть **NO**.

Вхідні дані

У першому рядку задано одне ціле число *k* — мінімальна кількість незалежних входжень рядка **TOILET**.

Другий рядок містить рядок *s* — повідомлення, надіслане Тойлет-мену. Рядок містить лише великі латинські символи.

Вихідні дані

У єдиному рядку виведіть **YES** або **NO** — відповідь на задачу.

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.005	1.031	Перегляд

Pull

Робота з командою:
оновили борду в Trello, обговорили проблематичні питання

Деякі встановлені розширення можуть впливати на якість виклику

Михальчук Антон Євгенійович (Ви покажете презентацію) | Звук презентації

Зупинити презентацію

artificial-intelligence-department / ai_programming_playground_2024

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

Files

main

Go to file

ai_11

ai_12

Epic 4: Stefan Shyika

Epic 5: Stefan Shyika

anton_mykhalechuk

bohdan_sonylo

bohdana_olynik

danylo_klymyshyn/epic_1

dmytro_tymchuk

eugenie_stefanovich

ivanna_smachylo

khyodrya_janiv

lesia_kashko

makym_perkhun

ai_programming_playground_2024 / ai_12

rtiska Merge pull request #173 from artificial-intelligence-department/epic_...

rtiska yesterday History

Name	Last commit message	Last commit date
...		
Epic 4: Stefan Shyika	final touches	last week
Epic 5: Stefan Shyika	minor changes	4 days ago
anton_mykhalechuk	Merge branch 'epic_2_practice_and_lesia_anton_mykhalechuk' of github.co...	2 days ago
bohdan_sonylo	Epic 2 - Bohdan Sonylo	3 days ago
bohdana_olynik	Epic 3 - Bohdana Olynik	3 weeks ago
danylo_klymyshyn/epic_1	Epic 1 - Danylo Klymyshyn	last week
dmytro_tymchuk	Epic 2 - Dmytro Tymchuk	2 weeks ago
eugenie_stefanovich	Add files via upload	2 days ago
ivanna_smachylo	Epic 3 - Ivanna Smachylo	3 days ago
...		

Бісюк Роман Васильович

Стефан

Ivanna Smachylo

Михальчук Антон Євгенійович

20:06 | nxs-qosq-fzf

Microphone

Camera

Screen

App Store

Google

Phone

Висновок: У ході роботи було вивчено основи роботи з файловою системою в C++: опрацьовано принципи обробки текстових і бінарних файлів, включаючи процеси запису, зчитування й редагування даних. Завдяки використанню різних типів рядкових змінних (`std::string` та `char*`) вдалося ознайомитися з різними підходами до зберігання й обробки текстових даних. Використання стандартної бібліотеки значно спростило роботу з файлами, дозволяючи зосередитися на вирішенні основних завдань.