Министерство и науки высшего образования

Пензенский государственный университет

Кафедра «Вычислительная техника»

Отчет

По лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных»

Выполнили студенты группы 21ВВ2:

Алёшина А.В.

Мубаряков А.А.

Приняли:

Юрова О.В.

Митрохин М.А.

Пенза 2022

Ход работы :

Была написана программа, вычисляющая сумму значений в каждом столбце (или строке) двумерного массива.

Были подключены библиотеки.

#include <iostream>

#include <windows.h>

#include <string>

Была написана программа, которая вычисляет разницу между максимальным и минимальным элементом. В этой же программе была int int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Задание 1,2\n");

const int max\_size = 10;

int array[max\_size];

for (int counter = 0; counter < max\_size; ++counter)

{

array[counter] = rand() % 200 - 100;

printf("%d ", array[counter]);

}

printf("\n");

int max\_element = 0;

int min\_element = 0;

for (int counter = 0; counter < max\_size; ++counter)

{

if (array[max\_element] < array[counter])

{

max\_element = counter;

}

if (array[min\_element] > array[counter])

{

min\_element = counter;

}

}

printf("Разница между %d и %d = %d\n", array[max\_element], array[min\_element], (array[max\_element] - array[min\_element]));

Был создан массив произвольного размера, вводимого с клавиатуры. А также была написана программа, вычисляющая сумму значений в каждой строке.

printf("\nЗадание 3,4\n");

int ROWS = 0, COLS = 0;

std::cout << "Введите количество строк: ";

std::cin >> ROWS;

std::cout << "Введите количество столбцов: ";

std::cin >> COLS;

int\*\* array2 = new int\*[ROWS];

for (int rows = 0; rows < ROWS; ++rows)

{

array2[rows] = new int[COLS];

}

for (int rows = 0; rows < ROWS; ++rows)

{

for (int cols = 0; cols < COLS; ++cols)

{

array2[rows][cols] = rand() % 200 - 100;

}

}

for (int rows = 0; rows < ROWS; ++rows)

{

for (int cols = 0; cols < COLS; ++cols)

{

printf("%4d ", array2[rows][cols]);

}

printf("\n");

}

for (int rows = 0; rows < ROWS; ++rows)

{

int sum\_value\_in\_row = 0;

for (int cols = 0; cols < COLS; ++cols)

{

sum\_value\_in\_row += array2[rows][cols];

}

printf("Сумма элементов строки [%d] = %d\n", rows, sum\_value\_in\_row);

}

for (int cols = 0; cols < COLS; ++cols)

{

int sum\_value\_in\_col = 0;

for (int rows = 0; rows < ROWS; ++rows)

{

sum\_value\_in\_col += array2[rows][cols];

}

printf("Сумма элементов столбца [%d] = %d\n", cols, sum\_value\_in\_col);

}

for (int counter = 0; counter < ROWS; ++counter)delete[] array2[counter];

delete[] array2;

Была написана программа, осуществляющая поиск среди структур student структуру с заданными параметрами.

struct Student

{

int number = 0;

std::string first\_name = "Имя";

std::string middle\_name = "Отчество";

std::string last\_name = "Фамилия";

int age = 0;

int year\_admission = 0;

int average\_score = 0;

Student\* next = nullptr;

Student()

{

std::cout << "Введите имя студента: ";

std::cin >> first\_name;

std::cout << "Введите фамилия студента: ";

std::cin >> last\_name;

std::cout << "Введите отчество студента: ";

std::cin >> middle\_name;

std::cout << "Введите возраст студента: ";

std::cin >> age;

std::cout << "Введите год поступления студента: ";

std::cin >> year\_admission;

std::cout << "Введите средний балл студента: ";

std::cin >> average\_score;

}

void output()

{

std::cout << "Студент номер " << number << ":" << std::endl;

std::cout << first\_name << std::endl << last\_name << std::endl << middle\_name << std::endl;

std::cout << "Возраст: " << age << std::endl;

std::cout << "Год поступления: " << year\_admission << std::endl;

std::cout << "Средний балл: " << average\_score << std::endl;

return;

}

};

struct List

{

Student\* first = nullptr;

Student\* last = nullptr;

bool is\_empty()

{

return first == nullptr;

}

void add\_back()

{

if (is\_empty())

{

first = new Student;

last = first;

first->number = 1;

return;

}

last->next = new Student;

int numberX = last->number;

last = last->next;

last->number = ++numberX;

return;

}

bool Search\_student(unsigned short int\* type, int number, std::string value)

{

bool found = 0;

bool anything = 0;

for (int counter = 0; counter < number; ++counter)

{

switch (type[counter])

{

case 1:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (pointer->first\_name == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

} break;

case 2:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (pointer->last\_name == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

} break;

case 3:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (pointer->middle\_name == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

} break;

case 4:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (std::to\_string(pointer->age) == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

} break;

case 5:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (std::to\_string(pointer->year\_admission) == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

} break;

case 6:

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

if (std::to\_string(pointer->average\_score) == value)

{

if (!anything)

{

std::cout << "По запросу \"" << value << "\" найдено:" << std::endl;

std::cout << "<------------------------------------------------------>" << std::endl;

anything = 1;

}

pointer->output();

std::cout << "<------------------------------------------------------>" << std::endl;

found = 1;

}

}

}

default: std::cout << "Неверный запрос" << std::endl;

}

}

return found;

}

void output()

{

for (Student\* pointer = first; pointer != nullptr; pointer = pointer->next)

{

pointer->output();

}

return;

}

~List()

{

if (is\_empty())

{

return;

}

Student\* pointer2 = nullptr;

for (Student\* pointer = first; pointer2 != nullptr; pointer = pointer2->next)

{

pointer2 = pointer->next;

delete pointer;

}

}

};

printf("Задание 5\n");

std::cout << "Заполните список студентов:\n";

List list;

for (bool end = 0; !end; std::cout << "\nЗакончить ввод?\n", std::cin >> end)

{

list.add\_back();

}

list.output();

unsigned short int\* type\_search = new unsigned short int[6]{0, 0, 0, 0, 0, 0};

unsigned short int\* buffer = new unsigned short int(-1);

int \*counter = new int(0);

std::cout << "По каким критериям произвести поиск?:\n1)Имя\n2)Фамилия\n3)Отчество\n4)Возраст\n5)Год поступления\n6)Средний балл\n0)Закончить ввод\n";

while (\*buffer != 0 && \*counter < 6)

{

std::cin >> \*buffer;

type\_search[\*counter] = \*buffer;

(\* counter)++;

}

delete buffer;

std::string request = "Empty Line";

std::cout << "Введите значение поиска : ";

std::cin >> request;

if (!list.Search\_student(type\_search, \*counter, request))

{

std::cout << "Ничего не найдено." << std::endl;

}

delete counter;

delete[] type\_search;

return 0;

}

Результаты работы программы:



Рисунок 1 Результат работы задания 1,2

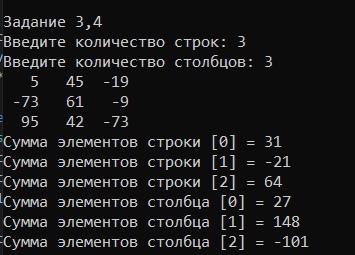


Рисунок 2 Результат работы Задания 3,4

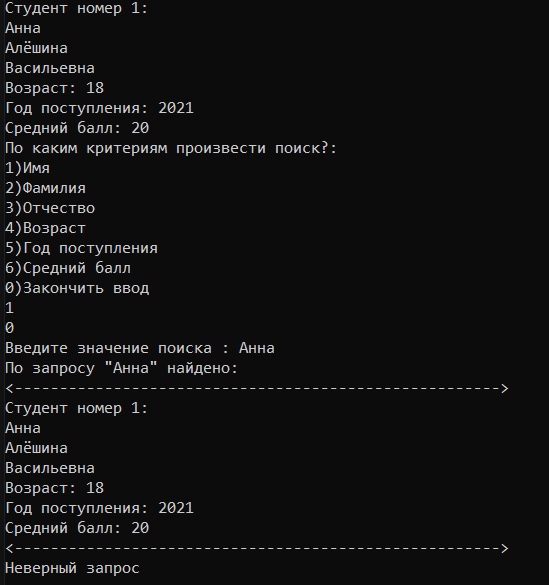
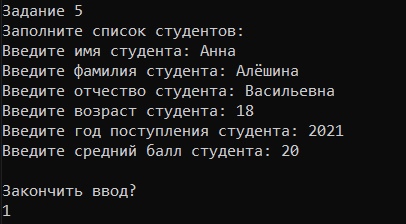


Рисунок 3 Результат работы задания 5

Вывод: данный алгоритм поиска информации по различным критериям показывает высокие скорости при сортировке.