**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Университетский политехнический колледж

Утверждаю

Зам. директора по УМР

Е.Г.Конакина

«\_\_\_» \_\_\_\_ 2018 г.

**ОТЧЕТ**

**по учебной практике УП.01.01**

**«Системное программирование»**

По профессиональному модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»

Специальность09.02.03. «Программирование в компьютерных системах»

Студента III курса32928/1 группы

Злобиной Полины Сергеевны

(Фамилия, имя, отчество)

Место прохождения практики: УПК, пр. Энгельса д.23

Период прохождения практики

с «17» сентября 2018 г. по «06» октября 2018 г.

Руководитель(и) практики:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Девятко Н.С.\_\_

(подпись) (расшифровка подписи)

Итоговая оценка по практике\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2018

Утверждаю

Зам. директора по УМР

Е.Г.Конакина

«\_\_\_» \_\_\_\_ 2018 г.

**Задание на учебную практику УП.01.01**

по профессиональному модулю

ПМ.01«Разработка программных модулей программного обеспечения для компьютерных систем»

Специальность09.02.03 «Программирование в компьютерных системах»

(код и наименование специальности)

Студенту \_\_\_3\_\_\_\_\_ курса \_\_32928/1\_\_\_ группы

Злобиной Полине Сергеевне

(фамилия, имя, отчество)

Место прохождения практики Университетский политехнический колледж

(наименование и адрес организации)

Период прохождения практики

с «17» сентября 2018 г.по «06» октября 2018 г.

**Виды работ, обязательные для выполнения**

Использование функций

Работа с указателями

Работа с файлами

Динамические структуры данных

Работа с файловой системой

Организация многопоточной обработки данных

**Индивидуальное задание** *(заполняется в случае необходимости дополнительных видов работ для решения практикоориентированных задач и т.д.)*

**ВАРИАНТ 5**

Задание выдал «17» сентября 2018 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Девятко Н.С

(подпись) (Ф.И.О.)

С заданиемознакомлен

«17» сентября 2018 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Злобина П.С

(подпись) (Ф.И.О. студента)

**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Университетский политехнический колледж

**ДНЕВНИК**

**прохождения учебной практики**

**«Системное программирование»**

По профессиональному модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»

Специальность09.02.03. «Программирование в компьютерных системах»

Студент(ка) III курса 32928/1 группы

Злобина Полина Сергеевна

Место прохождения практики УПК, пр. Энгельса д.23

Период прохождения практики

с «17» сентября 2018 г. по «06» октября 2018 г.

Руководитель(и) практики:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Девятко Н.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2018

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 17.09 | Разработка алгоритмов решения вычислительных задач. |  |
| 18.09 | Реализация программ по типовым алгоритмам. |  |
| 19.09 | Разработка спецификаций отдельных компонент программного обеспечения. Создание и использование функций. |  |
| 20.09 | Кодирование, отладка, тестирование, оптимизация программного модуля обработки простых данных с помощью функций программиста. |  |
| 21.09 | Организация динамического распределения памяти. Разработка функций для обработки, преобразования, сортировки данных с помощью адресной арифметики. |  |
| 22.09 | Кодирование, отладка, тестирование, оптимизация программного модуля обработки массивов с использованием указателей. |  |
| 24.09 | Разработка спецификаций функций анализа и обработки строчных величин. Создание статической библиотеки. |  |
| 25.09 | Кодирование, отладка, тестирование, оптимизация программного модуля обработки строк с использованием собственной библиотеки. |  |
| 26.09 | Разработка функций для работы с текстовыми файлами, организации ввода-вывода текстовой информации и ее хранения на внешних носителях. |  |
| 27.09 | Разработка программ для работы с файлами и папками. |  |
| 28.09 | Разработка программ обработки двоичных файлов. |  |
| 29.09 | Динамические структуры данных: односвязная очередь, стек, дерево. Разработка функций по организации работы с динамическими структурами данных. |  |
| 01.10 | Кодирование, отладка, тестирование, оптимизация программного модуля обработки динамических структур данных. |  |
| 02.10 | Структура каталога. Разработка программ для просмотра каталога диска, поиска файлов, определения и изменения атрибутов файлов. |  |
| 03.10 | Кодирование, отладка, тестирование, оптимизация программного модуля по работе с файловой системой. |  |
| 04.10 | Создание параллельных потоков и процессов обработки информации. |  |
| 05.10 | Синхронизация потоков и процессов. |  |
| 06.10 | Кодирование, отладка, тестирование, оптимизация программного модуля многопоточной обработки информации. |  |

**Оглавление**

[Тема 1: «Разработка программ по типовым алгоритмам» 5](#_Toc526164888)

[Задача 1.1. 5](#_Toc526164889)

[Задача 1.2. 8](#_Toc526164890)

[Тема 2: «Использование функций» 13](#_Toc526164891)

[Задача 2.1. 13](#_Toc526164892)

[Задача 2.2. 17](#_Toc526164893)

[Тема 3: «Указатели и динамическое распределение памяти» 21](#_Toc526164894)

[Задача 3.1. 21](#_Toc526164895)

[Задача 3.2. 27](#_Toc526164896)

[Тема 4: «Обработка строчных величин» 33](#_Toc526164897)

[Задача 4. 33](#_Toc526164898)

[Тема 5: «Работа с файлами» 37](#_Toc526164899)

[Задача 5.1. 37](#_Toc526164900)

[Задача 5.2. 44](#_Toc526164901)

[Тема 6: «Динамические структуры данных» 49](#_Toc526164902)

[Задача 6. 49](#_Toc526164903)

[Тема 7: «Работа с файловой системой» 53](#_Toc526164904)

[Задача 7. 53](#_Toc526164905)

[Тема 8: «Организация многопоточной обработки данных» 59](#_Toc526164906)

[Задача 8. 59](#_Toc526164907)

# **Тема 1: «Разработка программ по типовым алгоритмам»**

**Цель работы:** получить практические навыки по разработке программ по типовым алгоритмам.

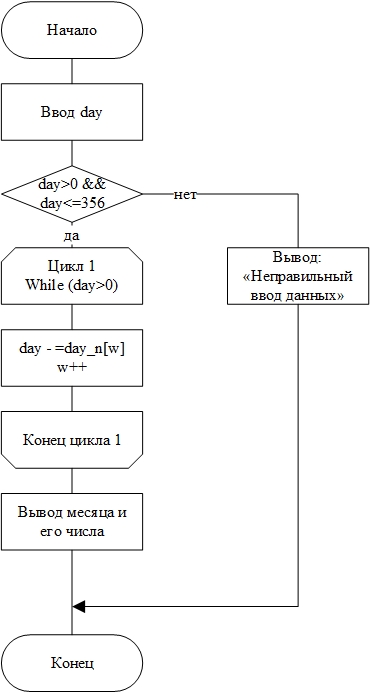
## **Задача 1.1.**

Составить алгоритм и программу, которая по номеру дня в году выводит число и месяц в общепринятой форме. Например, 33-ий день в году – 2 февраля.

**Математическая формулировка задачи:**

* Входные данные:day – тип int, переменная, предназначенная для ввода пользователем числа, в диапазоне от 1 до 365 (не високосный год) обозначающее день года.
* Выходные данные:w–тип int, переменная, предназначенная для подсчета номера месяца; При некорректном вводе данных, вывод сообщения предупреждающего об ошибке; При корректном вводе данных, вывод результата работы программы – месяца и его числа.
* Функциональные характеристики программы: программа должна обеспечивать ввод номера дня в году, контроль ввода, а затем выводить число и месяц в общепринятой форме.

**Схема основных функций:**



**Код программы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

usingnamespace std;

int\_tmain()

{

setlocale(LC\_ALL, "rus");

intday;//вводимыйдень

char \*month[] = { "Январь", "Февраль", "Март", "Апрель", "Май","Июнь", "Июль", "Август", "Сентябрь", "Октябрь", "Ноябрь", "Декабрь" };//массив названий месяцев

intday\_n[12] = { 31, 28, 31,30, 31, 30, 31, 31, 30, 31, 30, 31 };//массив кол-ва дней в каждом месяце

int w = 0;//переменная для хранения номера месяца

cout<<"Введите номер дня -> ";

cin>>day;

if ((day> 0) && (day<= 365))//Проверка условия ввода в диапазоне одного года

{

cout<<"По календарю, это: ";

while (day> 0)//цикл вычитания дней месяцев из введенного числа для счета номера месяца

{

day -= day\_n[w]; //счетчикдней

w++;//Увеличение номера месяца

}

cout<<month[w-1]<<", "<<(day\_n[w-1]+day)<<"-ечисло\n\n";

}

else

{

cout<<"Неправильный ввод данных!\n\n";

}

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  day (0;365], day>0, day<=365 | day = 3  day = 33  day = 345 | «Январь, 3-е число»  «Февраль, 2-е число»  «Декабрь, 11-е число»  При положительных значениях day в допустимом диапазоневыводятся месяц и число месяца, соответствуя дню года. (рис. 1, 2, 3) |
| 2 | Класс не допустимых значений  day∉ (0;365], day<0 | day = -45 | «Неправильный ввод данных!»  При отрицательных значениях day выводится ошибка ввода. (рис. 4) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 3 | Класс не допустимых значений  day∉(0;365], day>365 | day = 6000 | «Неправильный ввод данных!»  При превышающих значениях day в диапазоне, выводится ошибка ввода. (рис. 5) |
| 4 | Класс допустимых значений  day (0;365], day=1, day=365  граничные условия | day = 1  day = 365 | «Январь, 1-е число»  «Декабрь, 31-е число»  При крайнем диапазоне в пределах допустимых значениях, должен выводится 1-й месяц и 1-е число и 12-й месяц и 31-е число.  (рис. 6, 7) |



Рисунок 1



Рисунок 2



Рисунок 3



Рисунок 4



Рисунок 5



Рисунок 6



Рисунок 7

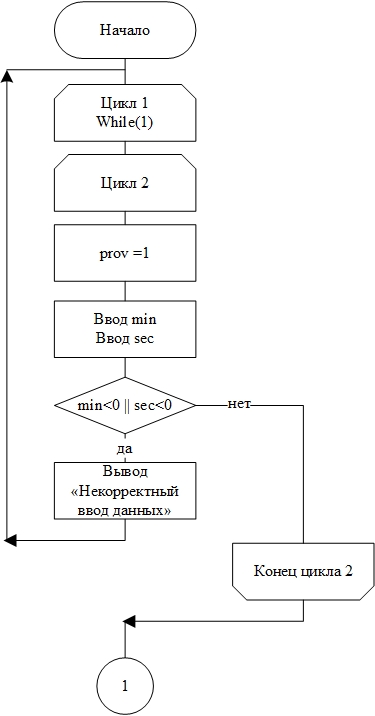
## **Задача 1.2.**

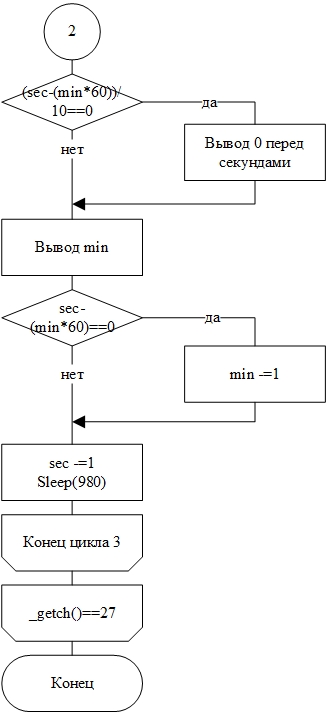
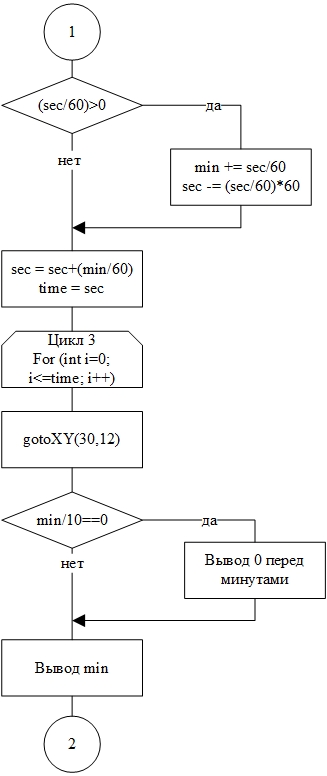
Составить алгоритм и программу, которая имитирует работу секундомера. Пользователь задает количество минут и секунд, затем программа показывает обратный отсчёт времени на чистом экране по центру, по окончании отсчёта выдаёт сообщение «Время истекло!», и возвращается в режим ввода. Выход из программы при нажатии клавиши Esc.

**Математическая формулировка задачи:**

* Входные данные:min – тип int, переменная предназначенная для ввода пользователем минут; sec–тип int, переменная предназначенная для ввода пользователем секунд.
* Выходные данные: по центру экрана программа выводит кол-во минут и секунд с обратным отсчетом в реальном времени. В случае если минут или секунд меньше 10, то впереди добавляется 0. При достижении 0 значения секунд программа должна выводить сообщение об окончании отсчета.
* Функциональные характеристики программы: программа должна обеспечивать ввод пользователю минут и секунд, а затем программа показывает обратный отсчёт времени на чистом экране по центру, по окончании отсчёта выдает сообщение «Время истекло!», и возвращаться в режим ввода. Также программа должна завершатся при нажатии клавиши Esc.

**Схема основных функций:**





**Код программы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<windows.h>

usingnamespace std;

voidgotoXY(intX, intY)

{

COORDcoord;

coord.X = X;

coord.Y = Y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

while (1)

{

system("cls");

intmin = 0, sec = 0;//min - переменная для хранения минут sec - переменная для хранения секунд

intprov = 1;//переменная для проверки корректности ввода

do//цикл для проверки ввода

{

prov = 1;

cout<<"\nВведите время\nМинуты -> ";

cin>>min;

cout<<"Секунды -> ";

cin>>sec;

if (min<0 || sec<0)//проверка ввода

{

prov = 0;

cout<<"Некорректный ввод данных! Введите повторно!";

}

} while (prov == 0);

if ((sec / 60) > 0)//проверка если секунд введено больше чем 60

{

min += sec / 60;//прибавление 60 секунд к минутам

sec -= (sec / 60) \* 60;//вычет из секунд минуты

}

sec = sec + (min \* 60);//перевод минут в секунды и прибавление их к другим секундам

inttime = sec;//переменная time служит дублером sec для цикла, ткsec меняется в цикле

for (inti = 0; i<= time; i++)

{

system("cls");

gotoXY(23, 11);//перевод строки в центр

cout<<"Оставшееся время: ";

if (min / 10 == 0)//добавление нуля перед минутами, если их меньше 10

{

cout<<"0";

}

cout<<min<<" мин. ";

if ((sec - (min \* 60)) / 10 == 0)//добавление нуля перед секундами, если их меньше 10

{

cout<<"0";

}

cout<< sec - (min \* 60) <<" сек.";

if (sec - (min \* 60) == 0)//убывание минут в отсчете

{

min -= 1;

}

sec -= 1;

Sleep(980);//остановка программы

}

cout<<"\nВремя истекло!\nНажмитеEnter для продолжения или Esc для выхода";

if (\_getch() == 27)//проверка хочет ли пользователь выйти

{exit(0);}}}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  min>=0, sec>=0 | min = 1, sec = 4  min = 20, sec = 15 | «Оставшееся время: 1 мин. 4 сек.»  «Оставшееся время: 4 сек.»  При положительных значениях minи sec, выводится обратный отсчет, соответствующий заданному времени. (рис. 1, 2) |
| 2 | Класс не допустимых значений  min>=0, sec<=0 | min = 3, sec = -6 | «Некорректный ввод данных! Введите повторно!»  При отрицательных значениях sec, выводится ошибка и программа простит ввести время повторно.  (рис. 3) |
| 3 | Класс не допустимых значений  min<=0, sec>=0 | min = -7, sec = 6 | «Некорректный ввод данных! Введите повторно!»  При отрицательных значениях min, выводится ошибка и программа простит ввести время повторно.  (рис. 4) |
| 4 | Класс допустимых значений  min>=0, sec>=0  граничные условия | min = 0, sec = 0 | «Оставшееся время: 0 сек.»  При положительных значениях minи sec, выводится обратный отсчет, соответствующий заданному времени. (рис. 5) |

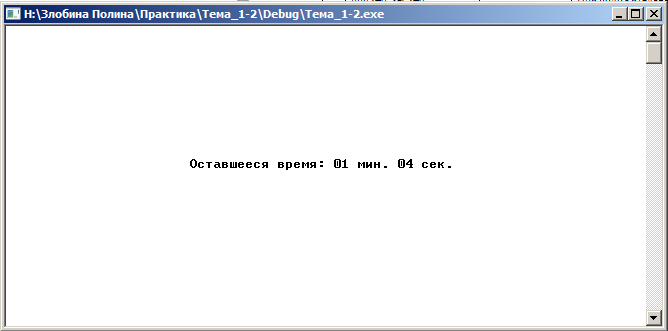


Рисунок 1

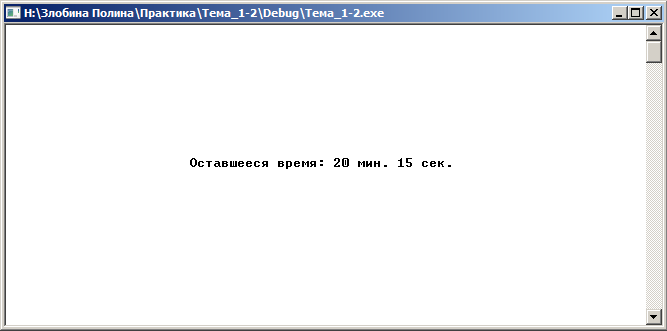


Рисунок 2

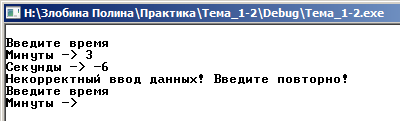


Рисунок 3

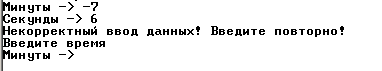


Рисунок 4

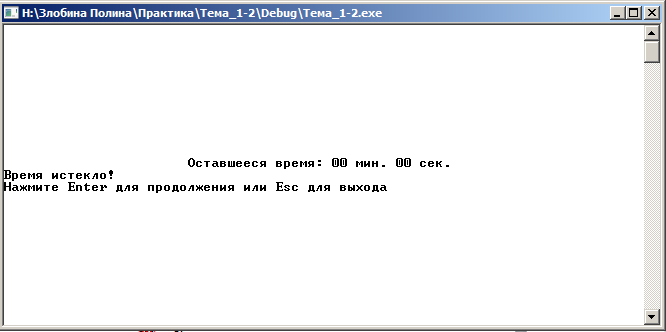


Рисунок 5

# **Тема 2: «Использование функций»**

**Цель работы:** получить практические навыки по использованию функций.

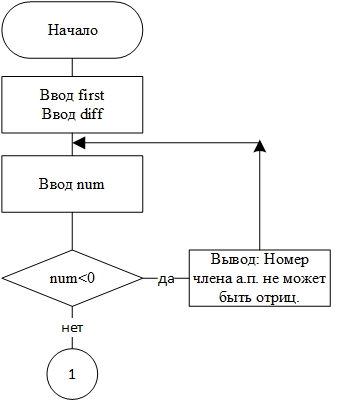
## **Задача 2.1.**

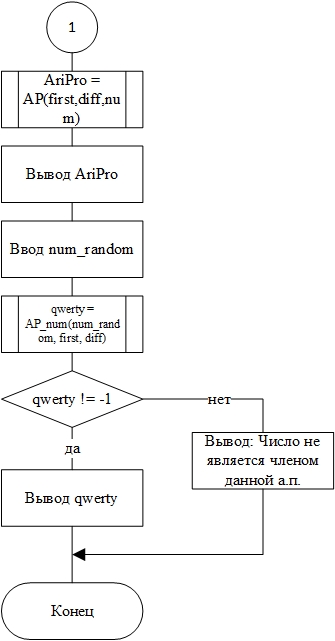
Разработать спецификации и написать функцию для вычисления значения любого члена арифметической прогрессии по заданному номеру, если известен первый член прогрессии и её разность, а также функцию, проверяющую, принадлежит ли заданное число этой прогрессии. Организовать вызов обеих функций для проверки.

**Математическая формулировка задачи:**

* Входные данные:first – тип int, переменная предназначенная для ввода пользователем первого члена арифметической прогрессии; diff–тип int, переменная предназначенная для ввода пользователем разности заданной арифметической прогрессии; num–тип int, переменная предназначенная для ввода пользователем номера искомого числа в заданной арифметической прогрессии.
* Выходные данные:AriPro – тип int, переменная предназначенная для вычисления введенного номера пользователем члена арифметической прогрессии; qwerty–тип int, переменная предназначенная для вычисления номера введенного пользователем числа в заданной арифметической прогрессии; при не нахождении введенного пользователем числа в заданной арифметической прогрессии выводится соответствующее сообщение.
* Функциональные характеристики программы: программа должна обеспечивать ввод первого члена арифметической прогрессии, разности, искомый номер члена и выполнять его вычисление, а также ввод пользовательского числа для нахождения оного в заданной арифметической прогрессии. Основные вычисления должны выполнятся в функции.

**Схема основных функций:**





**Код программы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

usingnamespace std;

int AP(intf, intd, intn)//функция для поиска члена АП

{

int sum = f;

for (inti = 1; i<n; i++)

{

sum += d;

}

return sum;

}

intAP\_num(intn\_r,intf, intd)//функциядляопределенияявл. личислочленомАП

{

double n;

n = ((double)(n\_r - f + d) / d);

if (n == round(n))//проверка целое ли число

{

return n;

}

elsereturn n=-1;

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

intfirst;//ПервыйчленАП

intdiff;//Разность АП

intnum;//номер члена АП

cout<<"Введите первый член арифметической прогресии -> ";

cin>>first;

cout<<"Введите разность а.п. -> ";

cin>>diff;

do

{

cout<<"Введите номер члена а.п., который вы хотите найти -> ";

cin>>num;

if (num< 0)

cout<<"\tНомер члена а.п. не может быть отрицательным или равным нулю!\n";

} while (num<0);

intAriPro = AP(first,diff,num);//вызываем ф-цию для высчитывания члена АП

cout<<"АПот ["<<num<<"] = "<<AriPro<<"\n\n\n";

cout<<"А теперь введите любое число -> ";

intnum\_random = 0;//пременная для вводимого на проверку числа

cin>>num\_random;

intqwerty = AP\_num(num\_random, first, diff);//вызов ф-ции для определения является ли число членом АП

if ( qwerty!= -1)

{

cout<<"Число "<<num\_random<<" является "<<qwerty<<"-м членом данной а.п.\n\n";

}

elsecout<<"Число не является членом данной а.п.\n\n";

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  first  diff  num>0  num\_random | first = 45  diff = 7  num = 5  num\_random = 477 | «АП от [5] = 73»  «Число не является членом данной а.п.»  При значениях переменных в допустимом диапазоне выводится член арифметической прогрессии под заданным номером, а также введенное num\_randomсравнивается с заданной арифметической прогрессией. (рис. 1) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 2 | Класс не допустимых значений  first  diff  num<0 | first = 45  diff = 7  num = -5 | «Номер члена а.п. не может быть отрицательным или равным нулю!»  При недопустимом значении num\_randomвыводится ошибка ввода. (рис. 2) |
| 3 | Класс допустимых значений  first  diff  num>0  num\_random  граничные условия | first = 388  diff = 27  num = 8  num\_random = 388 | «АП от [8] = 577»  «Число 388 является 1-м членом данной а.п.»  При значениях переменных в допустимом диапазоне выводится член арифметической прогрессии под заданным номером, а также введенное num\_randomсравнивается с заданной арифметической прогрессией. (рис. 3) |

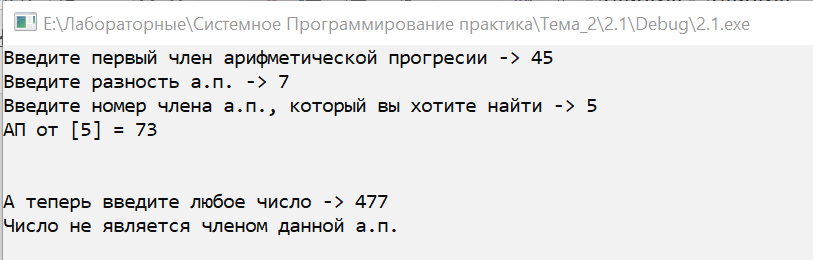


Рисунок 1

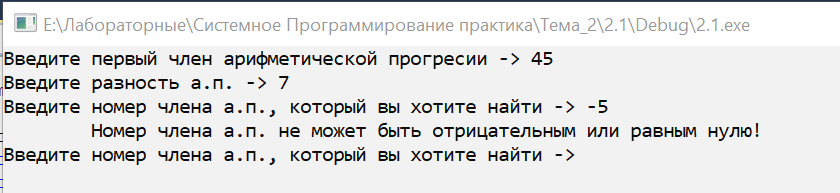


Рисунок 2

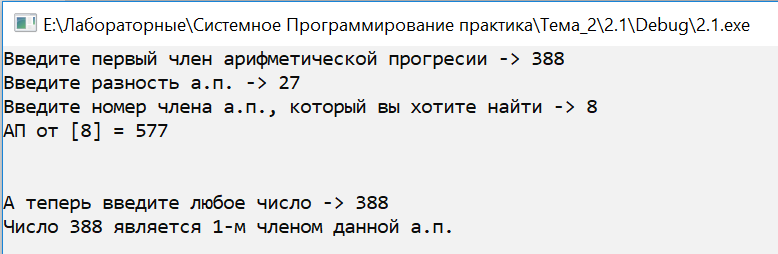


Рисунок 3

## **Задача 2.2.**

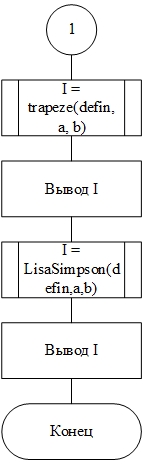
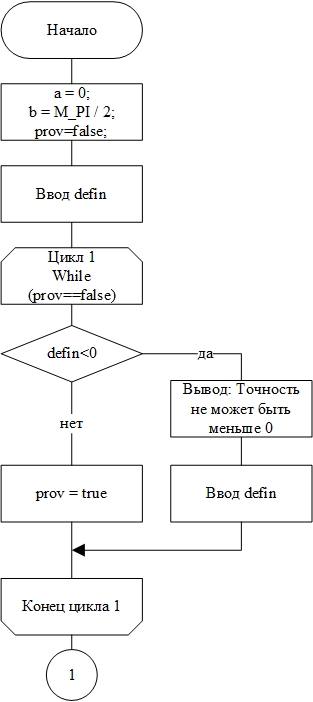
Разработать программу для вычисления интеграла с заданной точностью методом трапеций и методом Симпсона, оформив каждый способ в виде отдельной функции. Вывести на экран результаты интегрирования разными методами для сравнения. Подынтегральное выражение оформить в виде отдельной функции.



**Математическая формулировка задачи:**

* Входные данные: defin – тип int, переменная, предназначенная для ввода пользователем точности искомой площади интеграла.
* Выходные данные: вывод площади интеграла двумя разными способами, в виде двух отдельных функций. При вводе отрицательной точности, выдается соответствующее сообщение.
* Функциональные характеристики программы: программа должна обеспечивать ввод пользователем точности вычисления, а затем выводить сообщение с вычислениями интеграла двумя разными методами(трапеции и Симпсона), реализованные через две отдельные функции.

**Схема основных функций:**



**Код программы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

#define\_USE\_MATH\_DEFINES

#include<math.h>

usingnamespace std;

double function(doublex)

{

double I;

I = ((double)(sin(x)\*cos(x)) / (2 + pow(sin(x), 2) - pow(cos(x), 2)));

return I;

}

//методтрапеций

double trapeze(intdefin, doublea, doubleb)

{

double eps = 1;

double I = 0;

doubleI\_pred = 1;

for (inti = 0; i<defin; i++)//точность

{

eps = (double)eps / 10;

}

int n = 10;//кол-во повторений

while (abs(I\_pred - I) >eps&& n<10000000)//цикл будет выполнятся пока I не станет меньше eps

{

I\_pred = I;

double h = (double)(b - a) / n;

I = (function(a) + function(b)) / 2;//крайниезначения

for (int w = 1; w < n; w++)

{

I += function(a + w\*h);

}

I \*= h;

n \*= 2;

}

return I;

}

//методСимпсона

doubleLisaSimpson(intdefin, doublea, doubleb)

{

double eps = 1;

double I = 0;

doubleI\_pred = 1;

for (inti = 0; i<defin; i++)//точность

{

eps = (double)eps / 10;

}

int n = 10;//кол-во повторений

while (abs(I\_pred - I) >eps&& n<10000000)//цикл будет выполнятся пока I не станет меньше eps

{

I\_pred = I;

double h = (double)(b - a) / n;

I = function(a) + function(b);//крайниезначения

inttwo\_four;

for (int w = 1; w < n; w++)

{

if (w%2==0)//проверка для того чтобы менять число с 4 на 2, зависящее от номера прохода

{

two\_four = 2;

}

elsetwo\_four = 4;

I += function(a + w\*h)\*two\_four;

}

I \*= h/3;

n \*= 2;

}

return I;

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

double a = 0;

double b = (double)M\_PI / 2;

double I;

bool prov=false;

intdefin;

cout<<"Введите точность (простое число) -> ";

cin>>defin;

while (prov == false)//проверка на введенное число

{

if (defin< 0)

{

cout<<"Точность не может быть отрицательной или равной 0!"; prov = false;

cout<<"\nВведите точность -> "; cin>>defin;

}

else prov = true;

}

I = trapeze(defin, a, b);

cout<<"\nПлощадь заданного интеграла (метод трапеций) = "<< I <<"\n\n";

I = LisaSimpson(defin,a,b);

cout<<"\nПлощадь заданного интеграла (метод Симпсона) = "<< I <<"\n\n";

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  defin>0 | defin = 5 | «Площадь заданного интеграла (метод трапеций) = 0,27465»  «Площадь заданного интеграла (метод Симпсона) =0,274653»  При положительных значениях defin выводится вычисление интеграла по методу трапеций и Симпсона (рис. 1) |
| 2 | Класс не допустимых значений  defin<0 | defin = - 34 | «Точность не может быть отрицательной или равной 0!»  При отрицательных значениях defin, выводится ошибка и программа простит ввести точность повторно.  (рис. 2) |

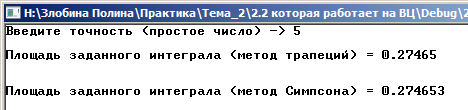


Рисунок 1

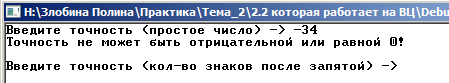


Рисунок 2

# **Тема 3: «Указатели и динамическое распределение памяти»**

**Цель работы:** получить практические навыки по использованию указателей и динамического распределения памяти.

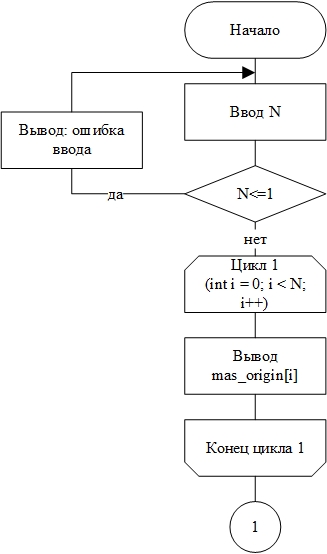
## **Задача 3.1.**

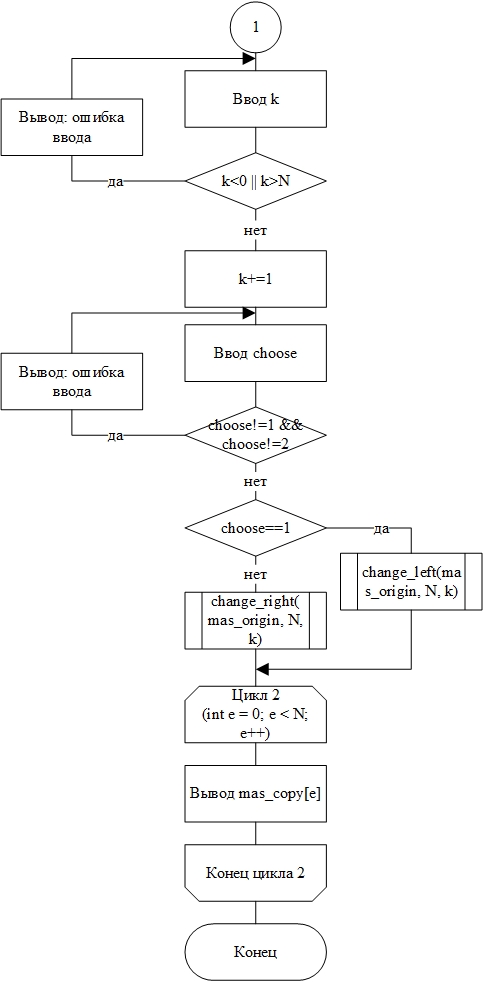
Динамически выделить память под N элементов вещественного типа float. Разработать алгоритм и программу обработки одномерного динамического массива. Осуществить циклический сдвиг элементов одномерного массива на k позиций.

**Математическая формулировка задачи:**

* Входные данные:N – тип int, переменная предназначенная для ввода пользователем размера исходного одномерного массива; k–тип int, переменная предназначенная для ввода пользователем числа сдвига исходного одномерного массива.
* Выходные данные:mas\_twin\_right–тип float, массив с осуществленным сдвигом на kпозиций вправо; mas\_twin\_left–тип float, массив с осуществленным сдвигом на kпозиций влево; При вводе недопустимой диапазоном размерности одномерного массива – выводится соответствующее сообщение; При вводе недопустимого кол-ва сдвига диапазоном – выводится соответствующее сообщение.
* Функциональные характеристики программы: программа должна обеспечивать ввод пользователем размерности массива и количество его сдвига, затем динамически выделять память под одномерный массив и осуществлять сдвиг этого массива на введенное количество элементов в правую и левую сторону.

**Схема основных функций:**





**Кодпрограммы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<stdlib.h>

Using namespace std;

void change\_right(float \*mas, intN, intk)

{

float \*p;

for (int j = 0; j <k; j++)

{

p = &mas[N - 1];

floatbuf = \*p;

while (p>=&mas[0])

{

\*p = \*(p-1);

p--;

}

p = mas;

\*p = buf;

}

}

voidchange\_left(float \*mas, intN, intk)

{

float \*p;

for (int j = 0; j <k; j++)

{

p = &mas[0];

floatbuf = \*p;//первыйэл

while (p <= &mas[N-1])

{

\*p = \*(p + 1);

p++;

}

p = &mas[N-1];

\*p = buf;

}

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

intN = 0, k = 0;//N - кол-воэлементов, k - позициисдвига

while (N <= 1)

{

cout<<"Введите кол-во элементов массива - > ";

cin>> N;

if (N <= 1)

{

cout<<"Количество эл. массива не может быть меньше двух!\n\n";

}

}

float \*mas\_origin = newfloat[N];//инициализ массива c указателем

srand(time(NULL));//чтобы в массиве были разные числа

for (inti = 0; i< N; i++)

{

mas\_origin[i] = (float)(rand()) / RAND\_MAX \* 500;// от -500 до 500

cout<<mas\_origin[i] <<" ";

}

while (k <= 0 || k>N)

{

cout<<"\n\nВведитечислосдвигамассива - > ";

cin>> k;

if (k <0 || k>N)

{

cout<<"Сдвиг не может быть отрицательными или быть больше массива!";

}

}

int choose = 0;

while (choose != 1 && choose != 2)

{

cout<<"\n\nВ какую сторону идет сдвиг? (1 - влево, 2 - вправо) -> ";

cin>> choose;

if (choose != 1 && choose != 2)

{

cout<<"Введите 1 или 2!";

}

}

if (choose == 1)

{

cout<<"\nСдвинутый массив влево:\n";

change\_left(mas\_origin, N, k);

}

else

{

cout<<"\nСдвинутый массив вправо:\n";

change\_right(mas\_origin, N, k);

}

for (int e = 0; e < N; e++)

{

cout<<mas\_origin[e] <<" ";

}

cout<<"\n\n";

delete[] mas\_origin;

system("pause");

cout<<endl<<endl<<endl;

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  N>1  k (0; N]  choose = {1,2} | N=5  k=2  choose = 1 | «479.003 37.1105 217.944 2.94504 235.42»  «Сдвинутый массив влево:  217.944 2.94504 235.42 479.003 37.1105»  При допустимых диапазоном значениях N и kвыводится сдвиг массива вправо и влево (рис. 1) |
| 2 | Класс не допустимых значений  N<1 | N= -66 | «Количество эл. массива не может быть меньше двух!»  При недопустимых диапазоном значениях N, выводится ошибка, и программа простит ввести размерность повторно.  (рис. 2) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 3 | Класс не допустимых значений  N>1  k>N | N= 6  k = 55 | «Сдвиг не может быть отрицательными или быть больше массива!»  При недопустимых диапазоном значениях k, выводится ошибка, и программа простит ввести значение сдвига повторно.  (рис. 3) |
| 4 | Класс не допустимых значений  N>1  k<0 | N= 4  k = - 45 | «Сдвиг не может быть отрицательными или быть больше массива!»  При недопустимых диапазоном значениях k, выводится ошибка, и программа простит ввести значение сдвига повторно.  (рис. 4) |
| 5 | Класс не допустимых значений  choose ≠ {1,2} | choose = 17 | «Введите 1 или 2!»  При недопустимых диапазоном значениях k, выводится ошибка, и программа простит ввести значение сдвига повторно.  (рис. 5) |

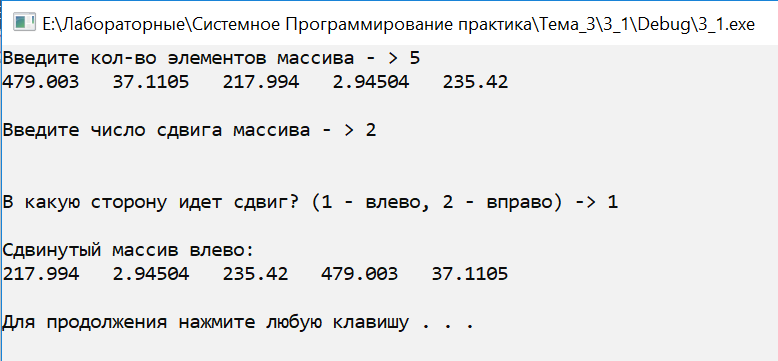


Рисунок 1

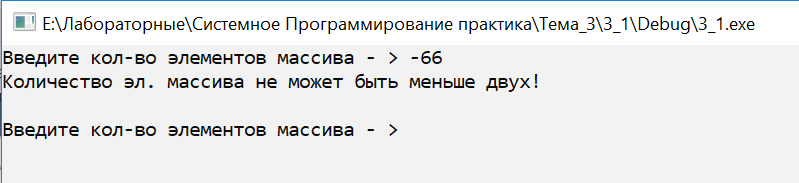


Рисунок 2

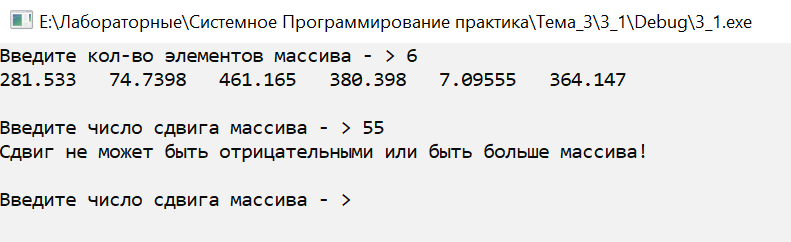


Рисунок 3

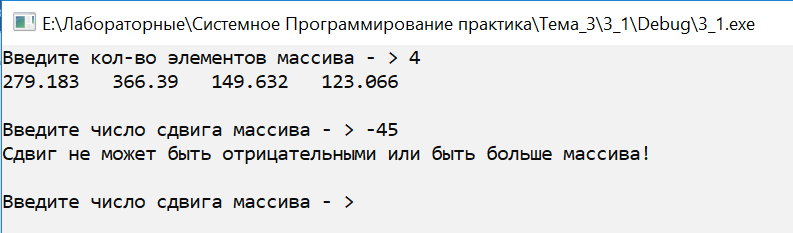


Рисунок 4

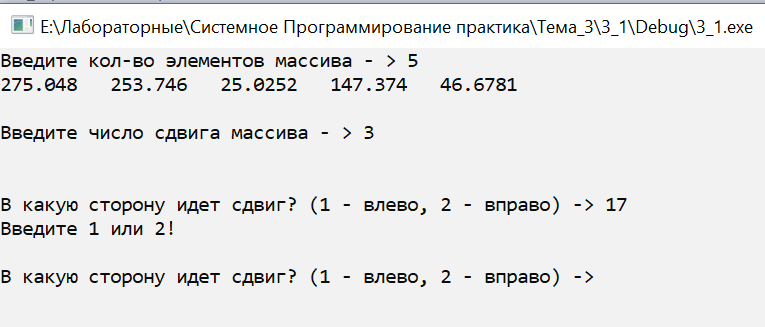


Рисунок 5

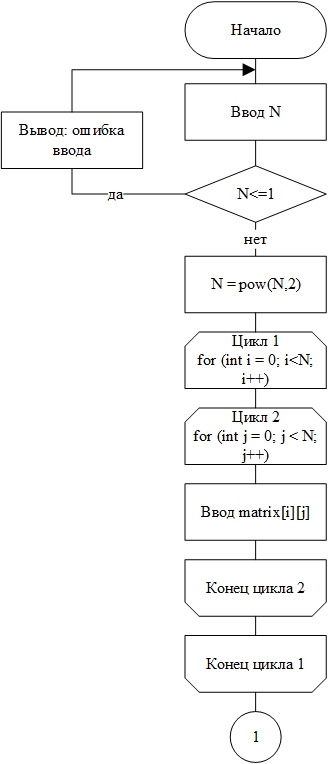
## **Задача 3.2.**

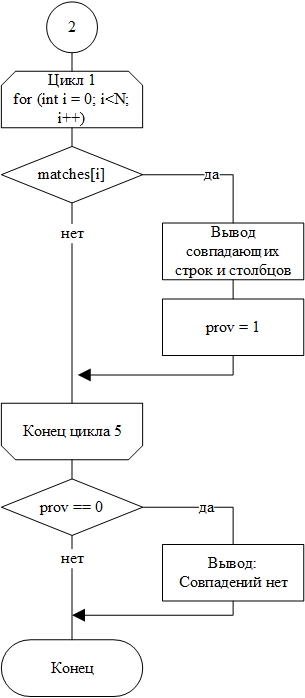
Динамически выделить память под N2 элементов целого типа int. Разработать алгоритм и программу обработки квадратной матрицы порядка N: найти в матрице такие номера k, что k –ая строка матрицы совпадает с k –ым столбцом.

**Математическая формулировка задачи:**

* Входные данные:N – тип int, переменная предназначенная для ввода пользователем размера двумерного массива; Ввод элементов двумерного массива.
* Выходные данные: вывод номеров строк и столбцов, совпадающих номерами и элементами, двумерного массива; при вводе недопустимой диапазоном размерности двумерного массива – выводится соответствующее сообщение;
* Функциональные характеристики программы: программа должна обеспечивать ввод пользователем размерности массива, а затем производить сравнение строк и столбцов по элементам и порядковым номерам строк и столбцов.

**Схема основных функций:**





**Код программы:**

#include"stdafx.h"

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<stdlib.h>

usingnamespace std;

voidDora\_the\_Explorer(int \*\*matrix, int \*matches, intN)

{

//mas[i, j] == \*(\*(mas+i)+j)

int \*p;

p = matches;

int flag;

for (inti = 0; i<N; i++)

{

flag = 0;

for (int j = 0; j <N; j++)

{

if (\*(\*(matrix + i) + j) == \*(\*(matrix + j) + i))

flag++;

}

if (flag == N)

\*p = 1;

else \*p = 0;

p++;

}

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

int N = 0;

while (N <= 1)

{

cout<<"Введите размерность кв. матрицы (N^2) - > ";

cin>> N;

if (N <= 1)

{

cout<<"Размерность матрицы должна быть больше одного!\n\n";

}

}

//int \*\*matrix = new int\*[N];//инициализ двумерной матрицы

N = pow(N, 2);

int \*\*matrix;

matrix = newint\*[N];

for (inti = 0; i< N; i++)

{

matrix[i] = newint[N];

}

srand(time(NULL));

cout<<"Матрица: \n";

for (inti = 0; i<N; i++)// циклпострокам

{

for (int j = 0; j < N; j++)// циклпостолбцам

{

cout<<"matrix["<<i + 1 <<","<< j + 1 <<"] = ";

cin>> matrix[i][j];

}

}

cout<<"\n";

for (inti = 0; i<N; i++)//выводматрицыдлянаглядности

{

for (int j = 0; j<N; j++)

{

cout<< matrix[i][j] <<" ";

}

cout<<"\n";

}

int \*matches = newint [N];

Dora\_the\_Explorer(matrix, matches, N);

int prov = 0;

for (inti = 0; i< N; i++)

{

if (matches[i])

{

cout<<"\nСовпадают строка и столбец под номером: "<< i+1;

prov = 1;

}

}

if (prov == 0)

{

cout<<"\nСовпадений не найдено! :(";

}

cout<<"\n";

delete[] matches;

delete[] matrix;

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  N>1 | N=2  matrix[1,1] = 3  matrix[1,2] = 8  matrix[1,3] = 7  matrix[1,4] = 6  matrix[2,1] = -27  matrix[2,2] = 18  matrix[2,3] = 14  matrix[2,4] = 96  matrix[3,1] = 7  matrix[3,2] = 14  matrix[3,3] = 0  matrix[3,4] = -3  matrix[4,1] = 666  matrix[4,2] = 13  matrix[4,3] = -3  matrix[4,4] = 1 | Найдено одно совпадение  При допустимых диапазоном значениях N выводится матрица, а также результат сравнения строк и столбцов (рис. 1) |
| 2 | Класс допустимых значений  N>1 | N=2  matrix[1,1] = 4  matrix[1,2] = 8  matrix[1,3] = 13  matrix[1,4] = 40  matrix[2,1] = 43  matrix[2,2] = -3  matrix[2,3] = -97  matrix[2,4] = -6  matrix[3,1] = 8  matrix[3,2] = 15  matrix[3,3] = -5  matrix[3,4] = -14  matrix[4,1] = 0  matrix[4,2] = 14  matrix[4,3] = -4  matrix[4,4] = 5 | «Совпадения не найдены»  При допустимых диапазоном значениях N выводится матрица, а также результат сравнения строк и столбцов (рис. 2) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 3 | Класс допустимых значений  N>1 | N=2  matrix[1,1] = 8  matrix[1,2] = 6  matrix[1,3] = -30  matrix[1,4] = -41  matrix[2,1] = 6  matrix[2,2] = 19  matrix[2,3] = 296  matrix[2,4] = -17  matrix[3,1] = -30  matrix[3,2] = 296  matrix[3,3] = 18  matrix[3,4] = 88  matrix[4,1] = -41  matrix[4,2] = -1  matrix[4,3] = 88  matrix[4,4] = 68 | Найдено несколько совпадений  При допустимых диапазоном значениях N выводится матрица, а также результат сравнения строк и столбцов (рис. 3) |
| 4 | Класс не допустимых значений  N<1 | N= -66 | «Размерность матрицы должна быть больше одного!»  При недопустимых диапазоном значенияхN, выводится ошибка, и программа простит ввести размерность повторно.  (рис. 4) |

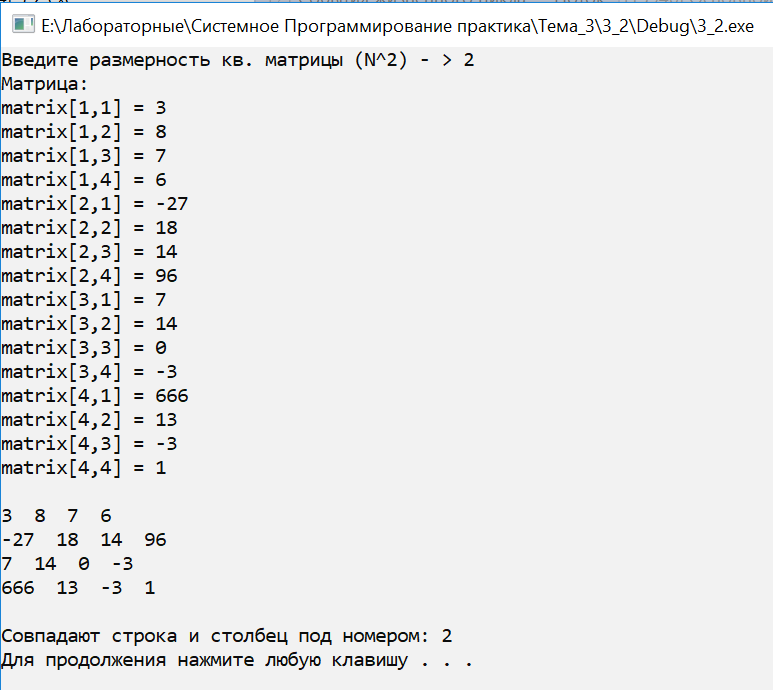


Рисунок 1

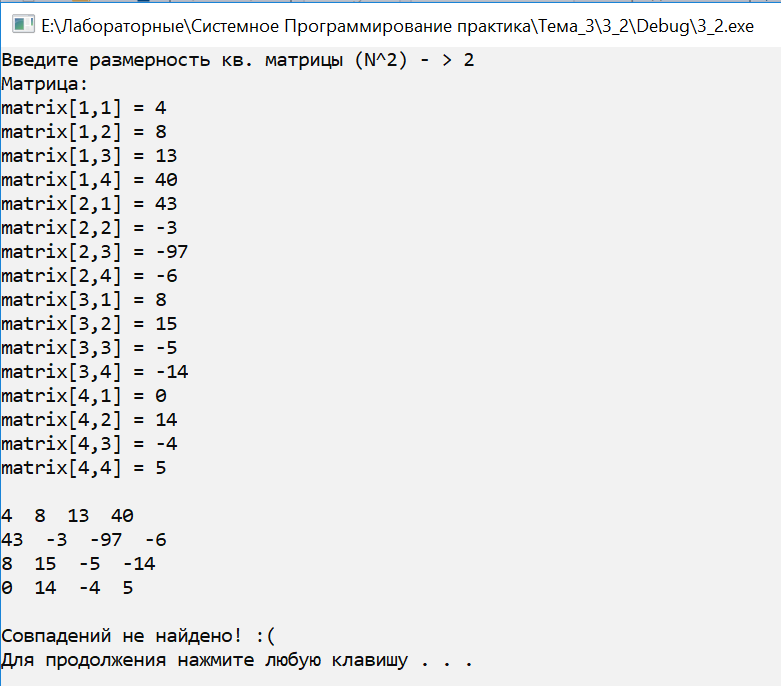


Рисунок 2

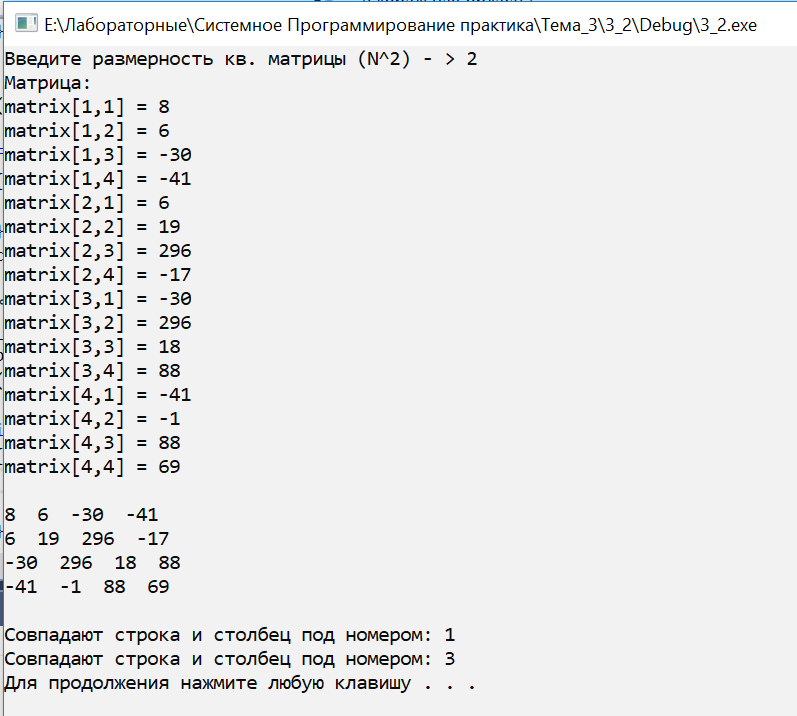


Рисунок 3

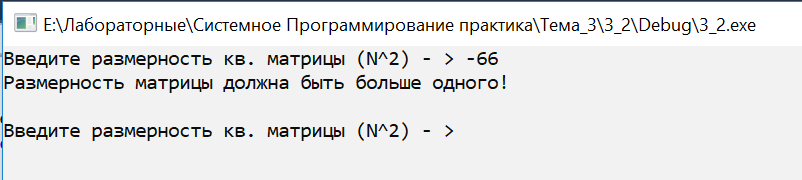


Рисунок 4

# **Тема 4: «Обработка строчных величин»**

**Цель работы:** получить практические навыки по обработке строчных величин.

## **Задача 4.**

* Разработать функцию, которая возвращает строку символов, которая получена из строки S1 путём удаления символов с позиции N1 до позиции N2.
* Разработать функцию, которая возвращает строку символов, которая получена из строки S1 путём удаления символов с позиции N1 до позиции N2 и вставки на это место строки S2.

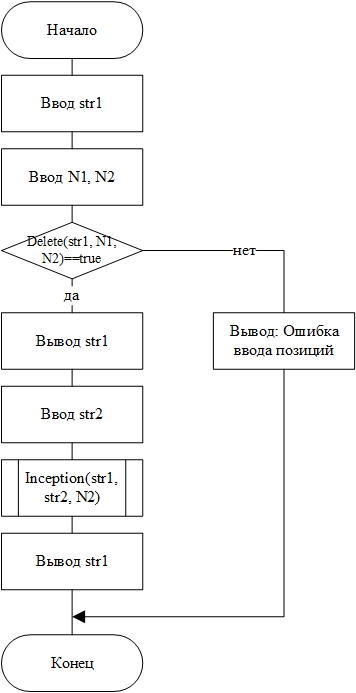
Все функции поместить в отдельном файле (библиотеке).

Разработать проект, подключающий собственную библиотеку, для проверки результатов вызова функций.

**Математическая формулировка задачи:**

* Входные данные: str1 – тип char, переменная, предназначенная для хранения введенной первой строки пользователем; str2 –тип string, переменная, предназначенная для хранения введенной второй строки пользователем; N1 –тип int, переменная, предназначенная для ввода начальной позиции; N2–тип int, переменная, предназначенная для ввода конечной позиции.
* Выходные данные: str1 – тип char, изменяемая в функциях строка; при некорректном вводе пользователем значениях N1 и N2 - выводится ошибка ввода.
* Функциональные характеристики: программа должна позволять пользователю вводить первую строку, затем удалять символы с позиции N1 до N2, также введенные пользователем. Далее пользователь вводит вторую строку, и программа подставляет вторую строку в первую строку на место удаленных символов.

**Схема основных функций:**



**Код программы:**

**MLi.cpp**

#include"stdafx.h"

#include<cstring>

#include"MLi.h"

int Delete(char \*str1, intN1, intN2)

{

int w = strlen(str1);

if (N1>w || N2<N1 || N1<0)

{

return 0;

}

else

{

char \*see = &str1[N1];

char \*see\_2 = &str1[N2 + 1];

for (inti = N1; i< w; i++)

{

\*see = \*see\_2;

see++; see\_2++;

}

return 1;

}

}

void Inception(char \*str1, char \*str2, intN1)

{

int w = strlen(str1) + strlen(str2);

int w\_2 = strlen(str1); w\_2 -= N1;

char \*see = &str1[N1];

char \*in = &str2[0];

char buff[1024] = { 0 };

for (int y = 0; y < w\_2; y++)

{

buff[y] = \*see; see++;

}

see = &str1[N1];

for (inti = N1; i< w; i++)

{

\*see = \*in;

see++; in++;

}

see = &str1[strlen(str1)];

for (int y = 0; y < w\_2; y++)

{

\*see = buff[y]; see++;

}

}

**MLi.h**

int Delete(char \*str1, intN1, intN2);

void Inception(char \*str1, char \*str2, intN1);

**Main**

#include"stdafx.h"

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<cstdlib>

#include<windows.h>

#include"MLi.h"//моябиблиотека

usingnamespace std;

int\_tmain()

{

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

constint size = 1024;

char str1[size] = { 0 };

charstr2[size] = { 0 };

int N1, N2;//N1 - начальная позиция удаления; N2 - последняя позиция удаления

cout<<"Введите строку -> \n\t";

cin.getline(str1, size);

cout<<"Введите с какого по какой символ удалить -> ";

cin>> N1 >> N2;

if (Delete(str1, N1, N2))//удаление символов на позициях с N1 по N2 и проверка на правильность ввода

{

cout<<endl<<"Новаястрока: "<< str1 <<endl<<endl;

cin.ignore();/\*Без этого не работает второй getline, тк он считает за введенный символ enter и проскакивает\*/

cout<<"Введите вторую строку(на английском) -> \n\t";

cin.getline(str2, size);

Inception(str1, str2, N1);

cout<<"Новаястрока: "<< str1 <<endl<<endl;

}

else { cout<<"Ошибка ввода позиций удаления!"; }

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  Строка str1  Строка str2  0<=N1<=размер строкиstr1  N2>N1 | str1 = «It’s dangerous to go alone! DANGER this!»  N1 = 28  N2 = 33  str2 = «Take» | «It’s dangerous to go alone! Take this!»  При допустимых значениях N1 и N2 удаляются символы с этих позиций в строке str1, а затем в первую строку добавляется строка str2. (рис 1.) |
| 2 | Класс не допустимых значений  Строка str1  N1>размер строки | str1 = «Houston, we have problems»  N1 = 384678942679 | «Ошибка ввода позиций удаления!»  При недопустимых диапазоном значенияхN1 и N2, выводится ошибка и программа завершается. (рис 2.) |
| 3 | Класс не допустимых значений  Строка str1  N2<N1 | str1 = «All your base belong to us!»  N1 = 5  N2 = 2 | «Ошибка ввода позиций удаления!»  При недопустимых диапазоном значенияхN1 и N2, выводится ошибка и программа завершается. (рис 3.) |

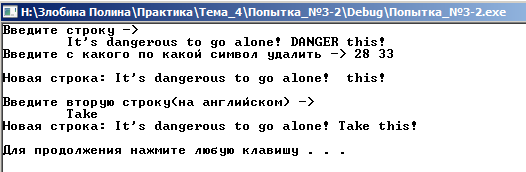


Рисунок 1

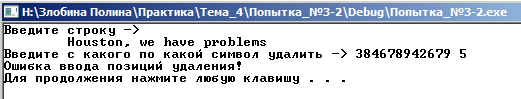


Рисунок 2

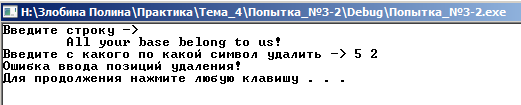


Рисунок 3

# **Тема 5: «Работа с файлами»**

**Цель работы:** получить практические навыки по работе с файлами.

## **Задача 5.1.**

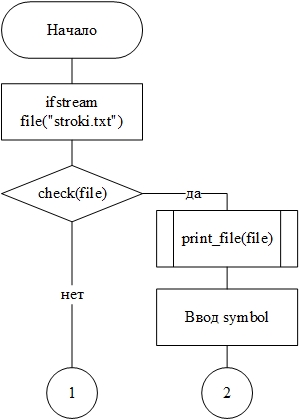
Создать в редакторе текстовый файл, внести в него несколько строк произвольного текста. Разработать программу, которая:

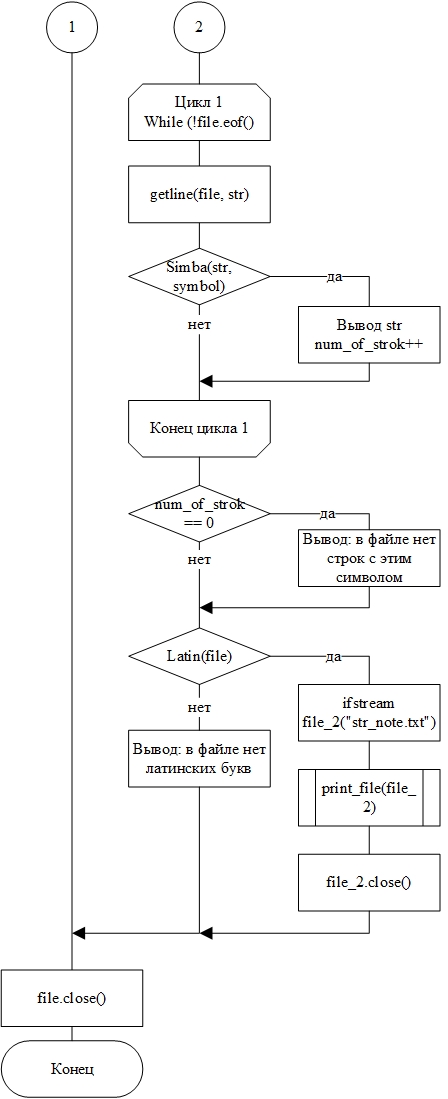
* выводит на экран строки файла, в которых встречается заданный символ
* формирует другой файл, в который переписывает строки исходного файла, содержащие латинские буквы, и указывает после каждой строки количество латинских букв в ней.

**Математическая формулировка задачи:**

* Входные данные: имя файла – в программе уже задано имя файла; symbol – тип char, переменная, предназначенная для хранения введенного пользователем символа, который необходимо найти в строке.
* Выходные данные: если файла не существует – программа выдает соответствующее сообщение; если файл пустой – программа выдает соответствующее сообщение; str–тип string, получаемая из файла строка.
* Функциональные характеристики: программа должна позволять пользователю вводить искомый символ, затем считывать из файла строки и проверять их на наличие искомого символа. Далее программа должна создавать файл, в который переписываются строки исходного файла содержащие латинские буквы и указывать после каждой строчки количество латинских букв в ней.

**Схема основных функций:**





**Код программы:**

#include"stdafx.h"

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<cstdlib>

#include<fstream>

#include<windows.h>

#include<string>

usingnamespace std;

bool check(ifstream&file)

{

if (!file.is\_open())

{

cout<<"(=>T\_T)=> Файл не найден"<<endl;

returnfalse;

}

elseif (file.peek() == EOF)

{

cout<<"Файлпуст<=(T\_T<=)"<<endl;

returnfalse;

}

else

{

returntrue;

}

}

voidprint\_file(ifstream&file)//выводфайла

{

stringstr;

while (!file.eof()) //цикл выполняется пока не закончится файл

{

getline(file, str);//получениестроки

cout<<"\t"<< str <<endl;//вывод

}

file.seekg(0);//возвращение в начало файла

}

boolSimba(stringstr, charsymbol)//проверка есть ли в считываемой строке нужный символ

{

char \*str\_ykaz = &str[0];//указатель на строку

while (\*str\_ykaz != '\0')

{

if (\*str\_ykaz == symbol)//c учетом регистра

{

returntrue;

}

str\_ykaz++;

}

returnfalse;

}

boolLatin(ifstream&file)//проверка принадлежит ли символ латнице и запись нового файла

{

ofstreamfile\_note("str\_note.txt");

file.seekg(0);//возвращение в начало файла

stringstr;

boolglobal = false;//проверка, запишется ли в файл хоть что либо

while (!file.eof())

{

getline(file, str);//получениестроки

intnum = 0;//счетчик кол-ва лат. букв

char \*str\_ykaz = &str[0];//указатель на строку

while (\*str\_ykaz != '\0')

{

if ((\*str\_ykaz>= 'A'&& \*str\_ykaz<= 'Z') || (\*str\_ykaz>= 'a'&& \*str\_ykaz<= 'z') || \*str\_ykaz == ' ')//c учетомрегистра

{

file\_note<< \*str\_ykaz; //записьвфайл

if (\*str\_ykaz != ' ')

{

num++;//подсчет кол-ва символов

global = true;

}

}

str\_ykaz++;

}

if (num != 0)//проверка чтобы не записывать пустую строку

{

file\_note<<"\t-> "<<num<<endl;//записывает в новый файл кол-во символов в строке

}

}

if (!global) return global;

elsefile\_note.close(); return global;

}

int\_tmain()

{

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

ifstream file("stroki.txt");

if (check(file))

{

string str;

intnum\_of\_strok = 0;

char symbol = ' ';

cout<<"Изначальный файл:"<<endl;

print\_file(file);//Вывод на экран содержимого файла

cout<<endl<<"Какой символ будем искать в файле? -> ";

cin>>symbol;

cout<<endl<<"Строки содержащие <"<<symbol<<"> символ:"<<endl;

while (!file.eof()) //цикл выполняется пока не закончится файл

{

getline(file, str);//получение строки

if (Simba(str, symbol))//проверка есть ли в считываемой строке нужный символ - задание 5.1.

{

cout<<"\t"<< str <<endl;//вывод

num\_of\_strok++;

}

}

if (num\_of\_strok == 0) cout<<"Нет строк с введенным символом!"<<endl;

cout<<endl<<"Строки ТОЛЬКО с латинскими буквами и их количеством:"<<endl;

if (Latin(file))

{

ifstream file\_2("str\_note.txt");

print\_file(file\_2);

cout<<"Записьфайлапроизведена!"<<endl;

file\_2.close();

}

elsecout<<"Вфайленетлатинскихбукв!"<<endl;

}

file.close();

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  symbol – любой  имя файла – «stroki.txt» | symbol = «W»  Файл содержит латинские буквы | «When the night#@^$ ends, it's not over»  «Weave right through to get closer КЛОСЕР!!!»  Программа считывает строку из файла и проверяет на совпадение с введенным символом. (рис.1) |
| 2 | Класс допустимых значений  symbol – любой  имя файла – «stroki.txt» | symbol = «Ф»  Файл содержит латинские буквы | «Этого символа не было в строках!»  Программа считывает строку из файла и проверяет на совпадение с введенным символом. (рис.2) |
| 3 | Класс допустимых значений  symbol – любой  имя файла – «stroki\_rus.txt» | symbol = «л»  Файл не содержит латинских букв | «В файле нет латинских букв!»  Программа считывает строки из исходного файла, проверяет на содержание латинских букв и записывает в новый файл. (рис.3) |
| 4 | Класс не допустимых значений  Имя файла – «not\_real.txt» | Файл не существует | «(=>T\_T)=> Файл не найден»  Программа проверяет, существует ли файл перед тем, как начать обработку строк.(рис.4) |
| 5 | Класс не допустимых значений  Имя файла – «empty.txt» | Файл не содержит строк | «Файл пуст <=(T\_T<=)»  Программа проверяет, пуст ли файл перед тем, как начать обработку строк. (рис.5) |

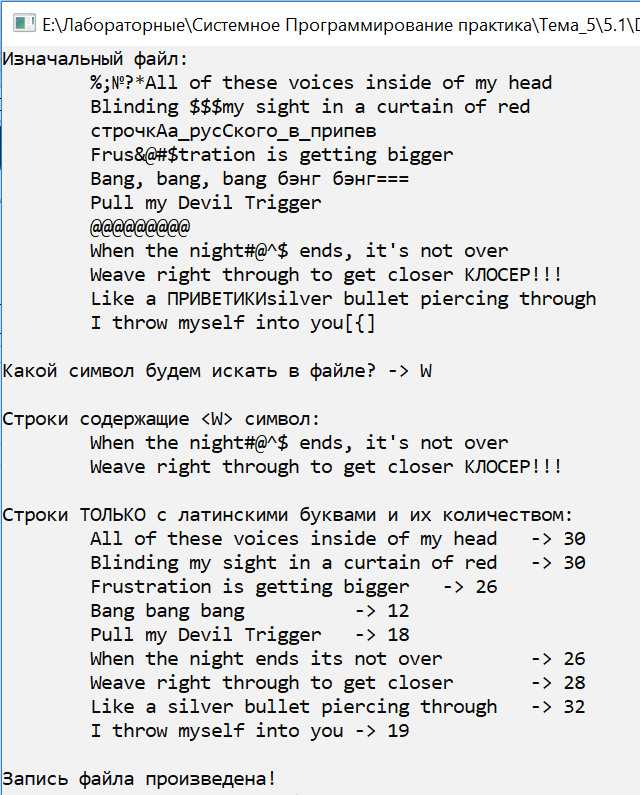


Рисунок 1

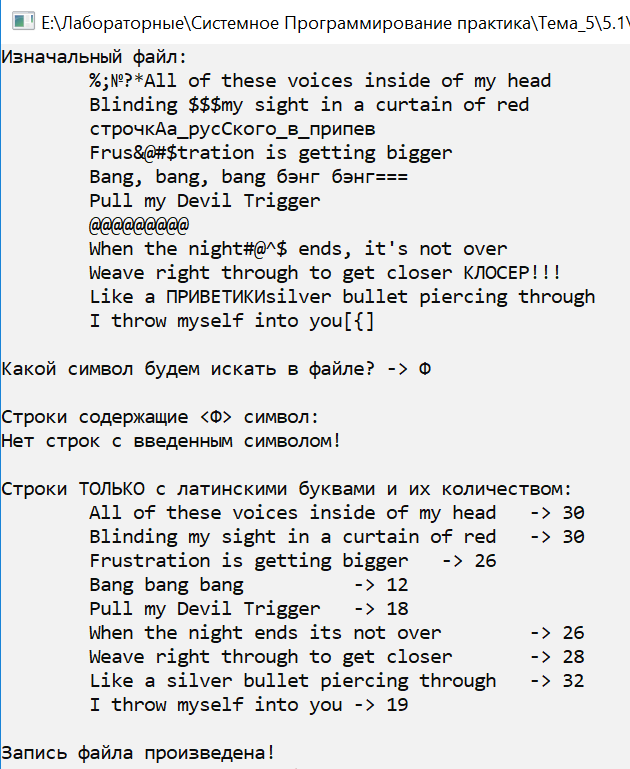


Рисунок 2

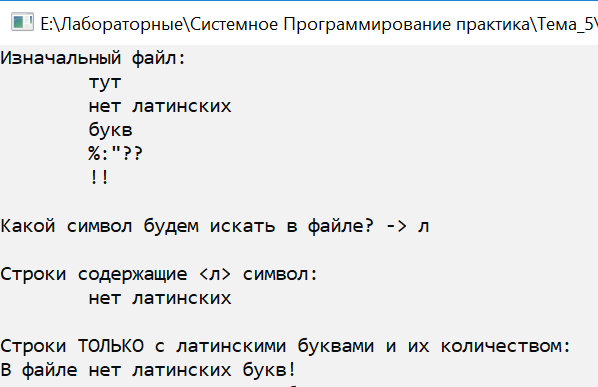


Рисунок 3

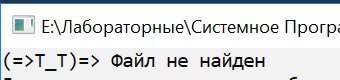


Рисунок 4

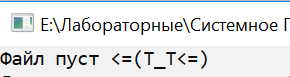


Рисунок 5

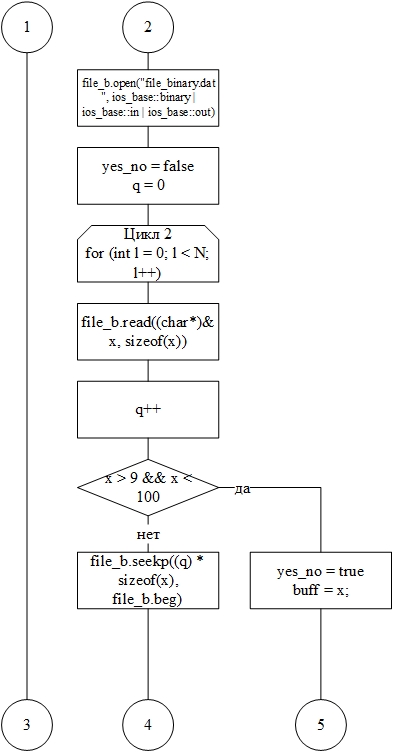
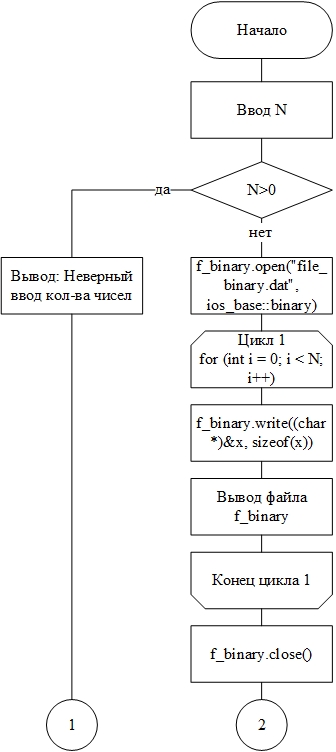
## **Задача 5.2.**

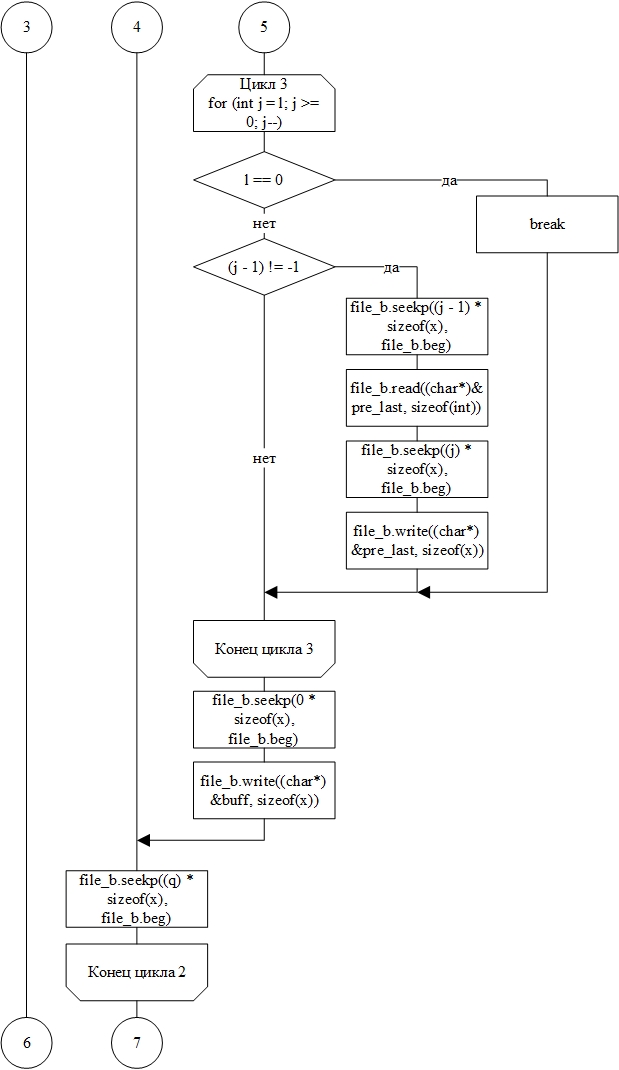
Разработать программу для заполнения двоичного файла случайными целыми числами в интервале [1; 500]. Найти компоненты файла, являющимися двузначными числами, и переместить их в начало файла. Массив не использовать.

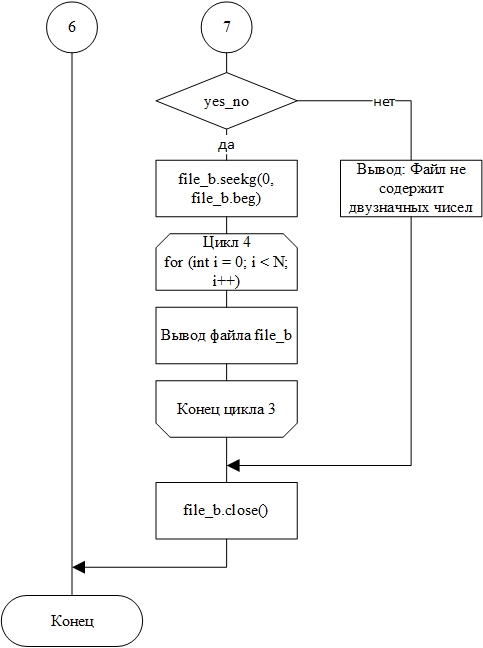
**Математическая формулировка задачи:**

* Входные данные: N – тип int, переменная предназначенная для хранения, введенного пользователем количества чисел в двоичной файле.
* Выходные данные: файл file\_binary.dat – измененный двоичный файл; при вводе недопустимого количества чисел – выводится соответствующее сообщение.
* Функциональные характеристики: программа должна позволять пользователю вводить количество элементов, создавать двоичный файл, записать туда введенное количество пользователем целых чисел, а затем найти и переместить в начало файла все двузначные числа.

**Схема основных функций:**







**Код программы:**

#include "stdafx.h"

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<cstdlib>

#include<ctime>

#include<fstream>

usingnamespace std;

int\_tmain()

{

setlocale(LC\_ALL, "rus");

int x, N;

cout<<"Введите кол-во чисел -> ";

cin>> N;

if (N > 0)

{

ofstreamf\_binary;

f\_binary.open("file\_binary.dat", ios\_base::binary);//открытиедлязаписи

srand(time(NULL));

cout<<"Исходныйфайл:"<<endl<<"\t";

for (int i = 0; i < N; i++)//заполнение случайными числами

{

x = 1 + rand() % 500;

cout<< x <<" ";

f\_binary.write((char\*)&x, sizeof(x));//записьвфайл

}

f\_binary.close();

fstreamfile\_b;//универсальноеоткрытие

file\_b.open("file\_binary.dat", ios\_base::binary | ios\_base::in | ios\_base::out);

int buff, pre\_last, q = 0;

boolyes\_no = false;//проверка есть ли вообше в файле двузначные

for (int l = 0; l < N; l++)//перепись

{

file\_b.read((char\*)&x, sizeof(x));//чтениеэлемента

q++;

if (x > 9 && x < 100)//попадение в диапазон

{

yes\_no = true;

buff = x;//хранение двузначного

for (int j = l; j >= 0; j--)//цикл для сдвига

{

if (l == 0) break;//если первый элемент и есть двузначное

if ((j - 1) != -1)//проверка чтобы не вылетело за границы файла

{

file\_b.seekp((j - 1) \* sizeof(x), file\_b.beg);//сдвиг на элемент стоящий перед двузначным

file\_b.read((char\*)&pre\_last, sizeof(int));//считываениеэлемента

file\_b.seekp((j) \* sizeof(x), file\_b.beg);//сдвигнадвузначное

file\_b.write((char\*)&pre\_last, sizeof(x));//перезаписьдвузначного

}

}

file\_b.seekp(0 \* sizeof(x), file\_b.beg);//сдвигвначало

file\_b.write((char\*)&buff, sizeof(x));//запись двузначного в начало файла

}

file\_b.seekp((q) \* sizeof(x), file\_b.beg);//возвращение на позицию

}

if (yes\_no)

{

file\_b.seekg(0, file\_b.beg);

cout<<endl<<"\nНовый файл (двузначные числа переставлены в начало): "<<endl<<"\t";

for (inti = 0; i< N; i++)

{

file\_b.read((char\*)&x, sizeof(x));

cout<< x <<" ";

}

cout<<endl;

}

elsecout<<endl<<endl<<"Файлнесодержитдвузначныхчисел!"<<endl<<endl;

file\_b.close();

}

elsecout<<endl<<"Неверныйвводкол-вачисел!"<<endl<<endl;

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  N>0  Имя файла – «file\_binary.dat» | N= «7»  Файл содержит двузначные числа | «Исходный файл:  8 69 145 17 136 33 230  Новый файл (двузначные числа переставлены в начало):  33 17 69 8 145 136 230»  При допустимом значении N, программа выводит исходный файл, а затем переставляет в нем числа так, чтобы двузначные были в начале файла. (рис. 1) |
| 2 | Класс допустимых значений  N>0  Имя файла – «file\_binary.dat» | N= «4»  Файл не содержит двузначные числа | «Файл не содержит двузначных чисел!»  Программа проверяет, если ли в файле двузначные числа и при их отсутствии выводит данное сообщение. (рис. 2) |
| 3 | Класс не допустимых значений  N<0 | N = «-44» | «Неверный ввод кол-ва чисел!»  При недопустимых значенияхN, выводится ошибка. (рис. 3) |

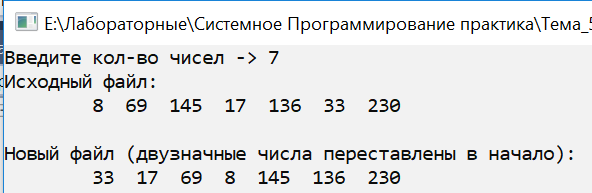


Рисунок1

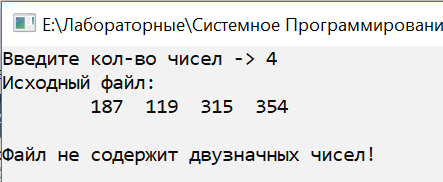


Рисунок 2

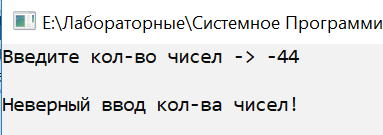


Рисунок 3

# **Тема 6: «Динамические структуры данных»**

**Цель работы:** получить практические навыки по работе с динамическими структурами данных.

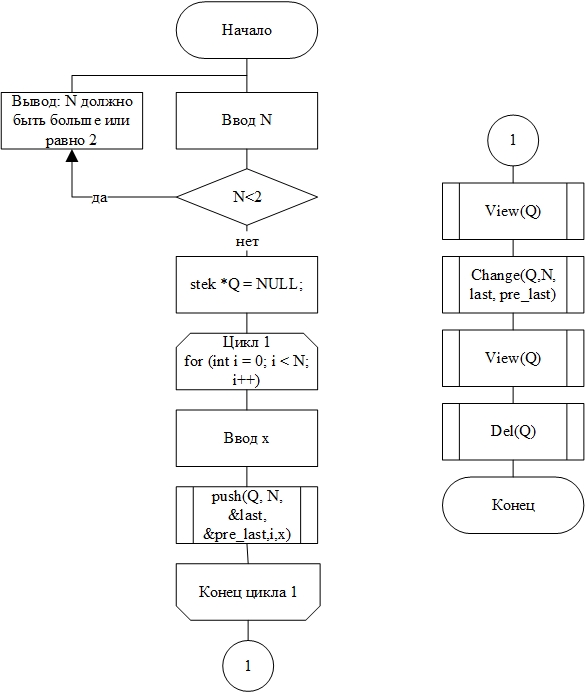
## **Задача 6.**

Составить программу обработки динамической структуры данных: поменять местами последний и предпоследний элементы стека Q, состоящего не менее чем из двух элементов.

**Математическая формулировка задачи:**

* Входные данные:N – тип int, переменная предназначенная для хранения, введенного пользователем количества чисел в динамической структуре данных; x–тип int, переменная предназначенная для ввода элементов динамической структуры.
* Выходные данные:Q – тип struct, динамическая структура данных; при вводе недопустимого количества чисел – выводится соответствующее сообщение.
* Функциональные характеристики: программа должна позволять пользователю вводить количество элементов, создавать динамическую структуру данных, затем находить последний и предпоследний ее элемент и менять их местами.

**Схема основных функций:**



**Код программы:**

#include"stdafx.h"

#include<conio.h>

#include<locale.h>

#include<iostream>

#include<ctime>

usingnamespace std;

structstek {

intvalue;

structstek \*next;// указательнаследующийэлементстека

};

void push(stek\* &next, intN, int \*last, int \*pre\_last, inti, intx)//заполнениестека

{

if (i == 0) \*last = x;//нахождение последнего

if (i == 1) \*pre\_last = x;//нахождение предпоследнего

stek \*t = newstek; // объявляем новую динамическую переменную типа stek

t->value = x;// записываем значение, которое помещается в стек

t->next = next; // связываем новый элемент стека с предыдущим

next = t; // новый элемент стека становится его вершиной

}

voidView(stek \*Q) //вывод стека на экран

{

stek \*t = Q;

while (t != NULL)

{

cout<< t->value <<" ";

t = t->next;

}

}

void Change(stek\*Q, intN, intlast, intpre\_last)//меняемместамипоследнийипредпоследн

{

stek \*t = Q;

int d = 0;

while (t != NULL)

{

if (N != 2)

{

t = t->next;

d++;

if (d == N - 2) t->value = last;

if (d == N - 1) t->value = pre\_last;

}

elsebreak;

}

if (N == 2)

{

t->value = last;

t = t->next;

t->value = pre\_last;

}

}

void Del(stek\* Q)

{

stek \*t = Q;

while (t != NULL)

{

t = Q->next;

deleteQ;

Q = t;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

int N = 0, last, pre\_last,x;

while (N < 2)//проверка на ввод

{

cout<<"\tКол-во элементов стека -> ";

cin>> N;

if (N < 2) cout<<"Кол-во элементов должно быть не менее 2-х"<<endl;

}

stek \*Q = NULL;//создание стека

cout<<"Введите элементы стека:"<<endl;

for (inti = 0; i< N; i++)

{

cout<<"Элемент ["<<i + 1 <<"] = ";

cin>> x;

push(Q, N, &last, &pre\_last,i,x);//заполнениестека

}

cout<<endl<<"\tЭлементыстека:"<<endl<<"\t\t";

View(Q);//вывод стека на экран

cout<<endl<<endl<<"\tПредпоследний элемент стека = "<<pre\_last<<endl<<"\tПоследний элемент стека = "<<last<<endl;

Change(Q, N, last, pre\_last);//заменаэлементов

cout<<endl<<"\tИзмененный стек:"<<endl<<"\t\t";

View(Q);//вывод стека на экран

Del(Q);//удалениестека

cout<<endl<<endl;

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  N>=2 | N= «7» | «Элементы стека:  280 242 41 36 295 94 63  Предпоследний элемент стека = 94  Последний элемент стека = 64  Измененный стек:  280 242 41 36 295 64 94»  При допустимом значении N, программа выводит исходный стек, а затем переставляет в нем предпоследний и последний элемент местами. (рис. 1) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 2 | Класс допустимых значений  N=2  граничные условия | N= «2» | «Элементы стека:  21 64  Предпоследний элемент стека = 21  Последний элемент стека = 64  Измененный стек:  64 21»  При допустимом значении N, программа выводит исходный стек, а затем переставляет в нем предпоследний и последний элемент местами. (рис. 2) |
| 3 | Класс не допустимых значений  N<2 | N = «1»  N = «-66» | «Кол-во элементов должно быть не менее 2-х»  «Кол-во элементов должно быть не менее 2-х»  При недопустимых значенияхN, выводится ошибка и программа просит ввести Nповторно. (рис. 3) |

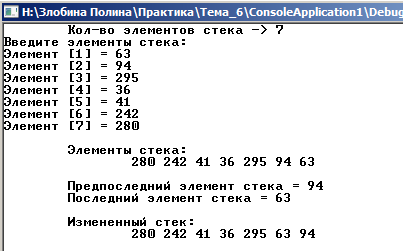


Рисунок 1

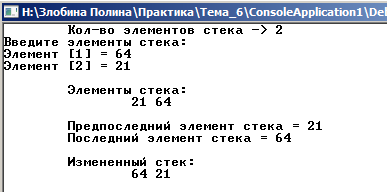


Рисунок 2

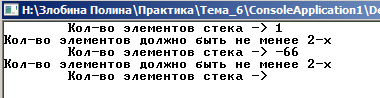


Рисунок 3

# **Тема 7: «Работа с файловой системой»**

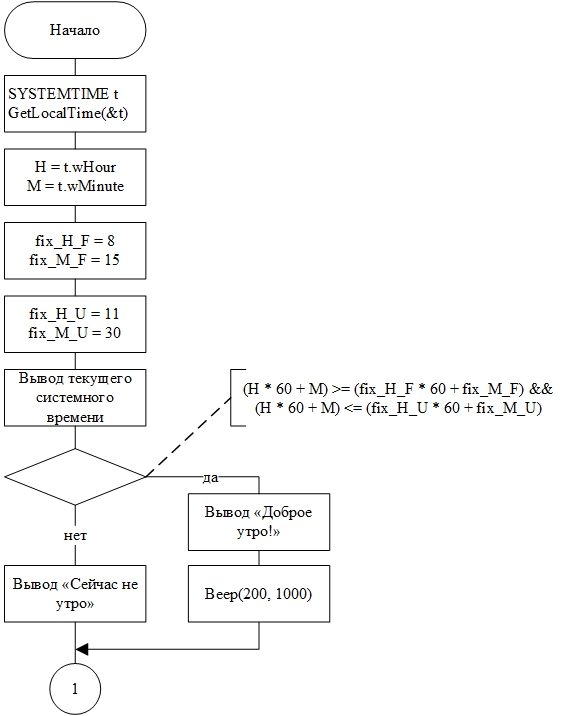
## **Задача 7.**

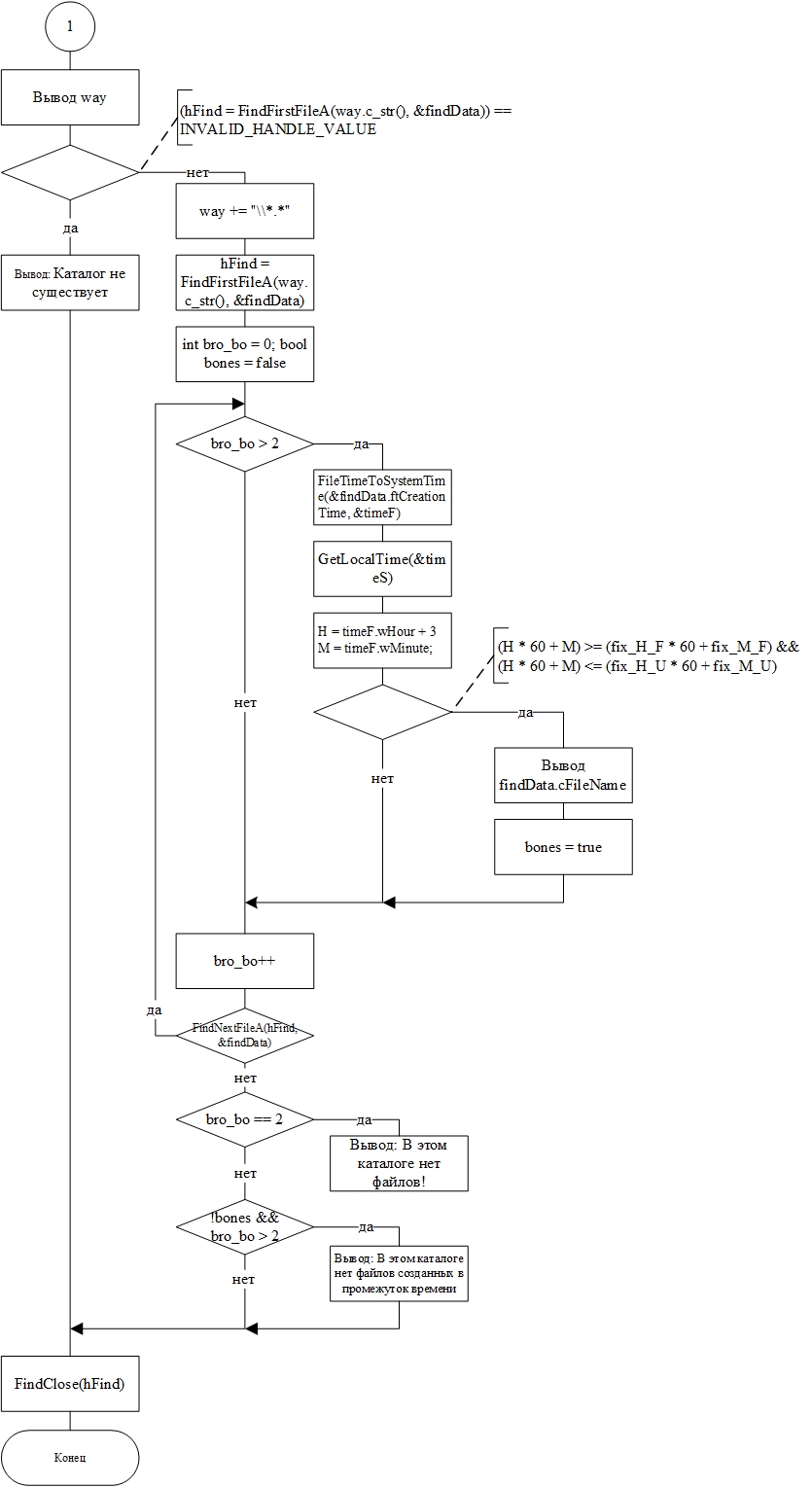
Составить программу по работе с файловой системой, которая позволяет узнать системное время, и, если время от 815 до 1130– выводит сообщение «Доброе утро!» и даёт звуковой сигнал, а также отыскивает в указанном каталоге файлы, созданные в это время.

**Математическая формулировка задачи:**

* Входные данные:way – тип string, переменна предназначенная для ввода пользователем пути к каталогу.
* Выходные данные:\_\_TIME\_\_- тип define, системное время; findData – тип WIN32\_FIND\_DATAA, имена файлов в каталоге; если пользователь попал в промежутки времени 815 до 1130 - выводится сообщение и звуковой сигнал; если каталога не существует – выводится соответствующее сообщение; если в каталоге нет файлов – выводится соответствующее сообщение.
* Функциональные характеристики: программа должна считывать, когда ее запустил пользователь и, если он попал в промежутки времени от 815 до 1130 - программа должна выводить сообщение и звуковой сигнал, а также выводить имена файлов созданных в этом промежутке, считываемых из введенном пользователем каталоге.

**Схема основных функций:**





**Код программы:**

#include"stdafx.h"

#define\_CRT\_SECURE\_NO\_WARNINGS

#pragmawarning(disable:4996)

#include<cstdlib>

#include<iostream>

#include<string>

#include<windows.h>

#include<iomanip>

#include<map>

#include<cstdio>

usingnamespacestd;

/\*Проект -> Свойства -> C/C++ -> Препроцессор -> Определения препроцессора -><Изменить>

-> \_CRT\_SECURE\_NO\_WARNINGS\*/

int main()

{

setlocale(LC\_ALL, "rus");

SYSTEMTIME t;

GetLocalTime(&t);

int H = t.wHour, M = t.wMinute;

intfix\_H\_F = 8, fix\_M\_F = 15;//час и время начала временых границ

intfix\_H\_U = 11, fix\_M\_U = 30;//час и время конца временых границ

//int H = 8, M = 50;

cout<<"Текущее системное время: "<<\_\_TIME\_\_<<endl;

//cout<< "Сейчас " <<t.wHour<< " ч. и " <<t.wMinute<< " мин." <<endl;

if ((H \* 60 + M) >= (fix\_H\_F \* 60 + fix\_M\_F) && (H \* 60 + M) <= (fix\_H\_U \* 60 + fix\_M\_U))

{

cout<<"Доброе утро! ;)"<<endl;

Beep(200, 1000);//звуковой сигнал

}

elsecout<<"Сейчас не утро :3"<<endl;

string way;//путькфайлам

WIN32\_FIND\_DATAAfindData;

HANDLEhFind;

cout<<endl<<"Путьккаталогу: ";

cin>> way;

if ((hFind = FindFirstFileA(way.c\_str(), &findData)) == INVALID\_HANDLE\_VALUE)//есликаталоганет

{

cerr<<"\n\n\tКаталог не существует\n\n"<<endl;

system("pause");

return 1;

}

way+="\\\*.\*";//Путь и имена файлов с разным расширением

hFind = FindFirstFileA(way.c\_str(), &findData);

SYSTEMTIMEtimeF, timeS;

cout<<endl<<"Файлывданномкаталоге|Времясоздания"<<endl;

intbro\_bo = 0; bool bones= false;

do

{

if (bro\_bo> 2)//костыль чтобы не выводить "." и ".." каталоги

{

FileTimeToSystemTime(&findData.ftCreationTime, &timeF);

GetLocalTime(&timeS);

H = timeF.wHour + 3;//Тк. время выводится по UTC, до нашего часового пояса +3 (файлы созданы по UTC)

M = timeF.wMinute;

if ((H \* 60 + M) >= (fix\_H\_F \* 60 + fix\_M\_F) && (H \* 60 + M) <= (fix\_H\_U \* 60 + fix\_M\_U))

{

cout<<findData.cFileName<<"\t-> "<< H <<":"<< M <<endl;

bones = true;

}

}

bro\_bo++;

} while (FindNextFileA(hFind, &findData));

if (bro\_bo == 2) cout<<"В этом каталоге нет файлов!"<<endl;

if (!bones&&bro\_bo>2) cout<<"В этом каталоге нет файлов, созданных в промежуток времени от 8.15 до 11.30"<<endl;

FindClose(hFind);

cout<<endl<<endl<<endl;

system("pause");

}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  Правильно написанный, существующий каталог  Программа запущена в промежутке времени от 815 до 1130часов | way= «D:\For\_task\_7» | Программа выведет на экран системное время, сообщение «Доброе утро!» и подаст звуковой сигнал, имена всех файлов каталога с расширением и временем создания в промежутке от 815 до 1130часов. (рис. 1) |
| 2 | Класс допустимых значений  Правильно написанный, существующий каталог  Программа запущена в не в промежутке времени от 815 до 1130часов | way= «D:\For\_task\_7» | Программа выведет на экран системное время, сообщение «Сейчас не утро :3», имена всех файлов каталога с расширением и временем создания в промежутке от 815 до 1130 часов. (рис. 2) |
| 3 | Класс допустимых значений  Правильно написанный, существующий каталог в котором нет файлов | way= «D:\empty» | «В этом каталоге нет файлов!»  Программа выведет на экран системное время, сообщение в зависимости от промежутка времени, а также сообщение о том, что в каталоге нет файлов.(рис. 3) |

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 4 | Класс допустимых значений  Правильно написанный, существующий каталог в котором нет файлов, созданных в промежутке времени от 815 до 1130часов | way= «D:\For\_task\_7» | «В этом каталоге нет файлов, созданных в промежуток времени от 8.15 до 11.30»  Программа выведет на экран системное время, сообщение в зависимости от промежутка времени, а также сообщение о том, что в каталоге нет файлов, созданных в промежутке от 815 до 1130 часов. (рис. 4) |
| 5 | Класс недопустимых значений  Несуществующий каталог | way = «Z:\HAHAH» | «Каталог не существует»  Программа выведет на экран системное время, сообщение в зависимости от промежутка времени, а также сообщение о том, что каталог не существует. (рис. 5) |

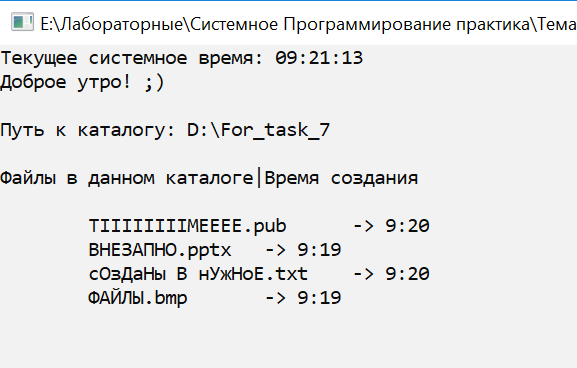


Рисунок 1

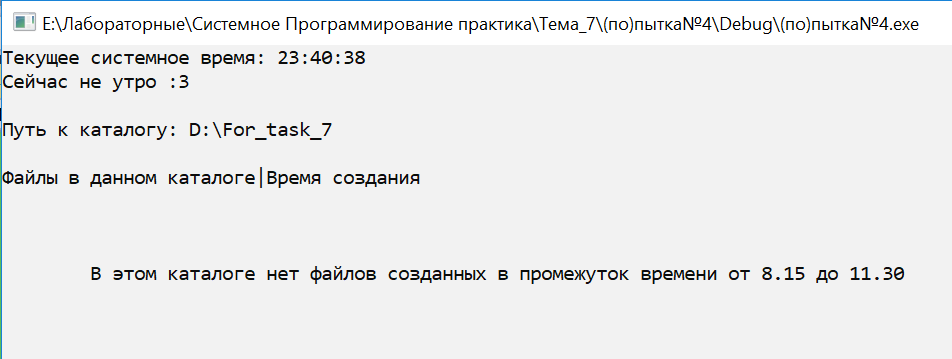


Рисунок 2

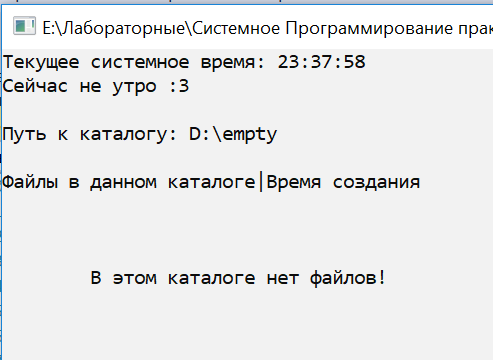


Рисунок 3

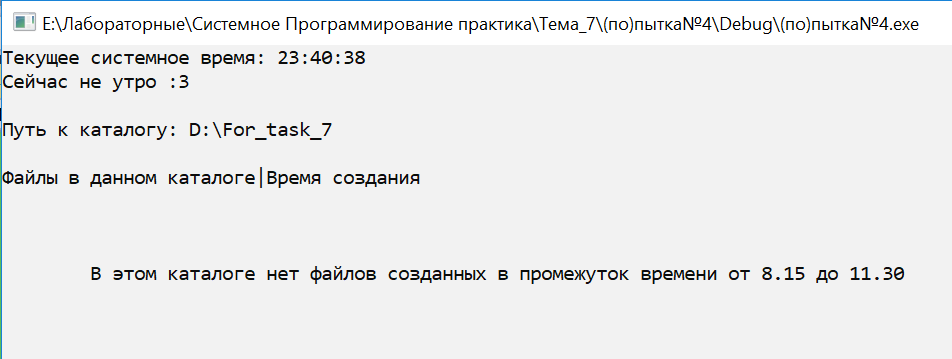


Рисунок 4

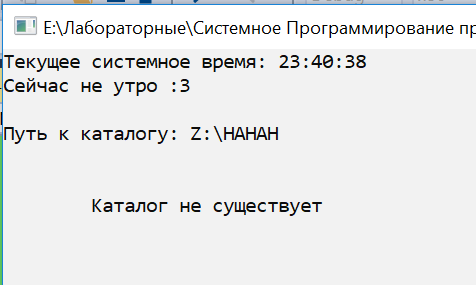


Рисунок 5

# **Тема 8: «Организация многопоточной обработки данных»**

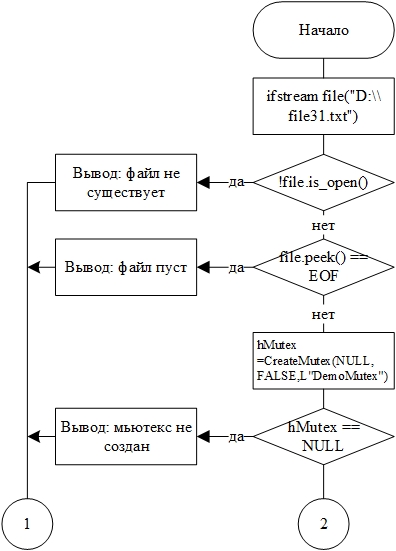
## **Задача 8.**

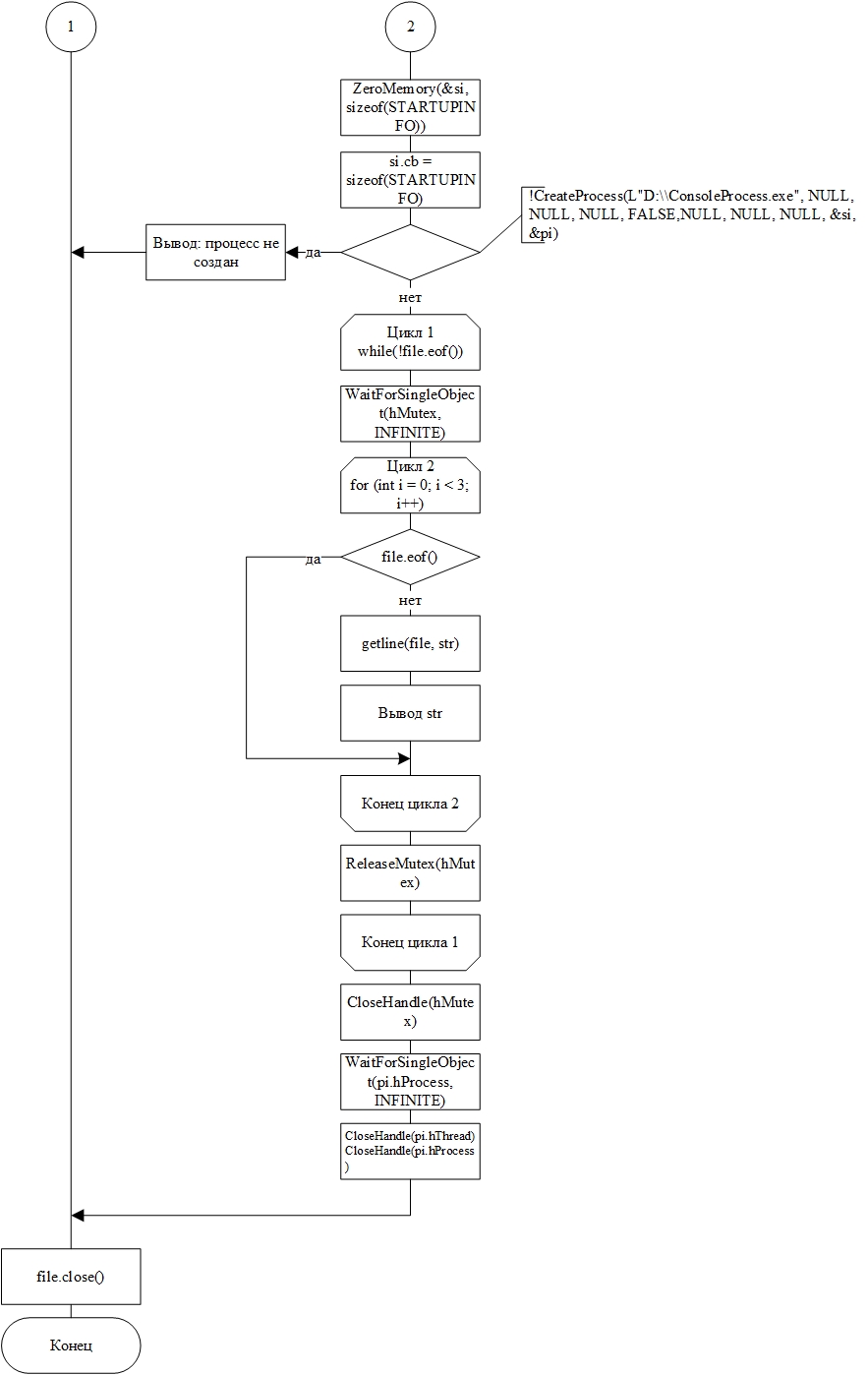
Напишите программу, которая создает процесс. Используйте атрибуты по умолчанию. Родительский и вновь созданный процесс должны распечатать строки текста из файла так, чтобы вывод родительского и дочернего процесса был синхронизован: сначала родительский процесс выводил бы три строки, затем дочерний пять строк, затем родительский следующие три строки и т.д. Используйте мьютексы.

**Математическая формулировка задачи:**

* Входные данные: имя файла – в программе уже задано имя файла.
* Выходные данные: str – тип string, вывод строки файла родительским или дочерним процессом; если файла не существует – программа выводит сообщение об этом; если файл пуст – программа выводит сообщение об этом.
* Функциональные характеристики: программа должна распечатать строки текста из файла так, чтобы вывод родительского и дочернего процесса был синхронизован: сначала родительский процесс выводил три строки, затем дочерний пять строк и т.д.

**Схема основных функций:**





**Код программы:**

**Основная программа**

#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include<fstream>

#include<string>

using namespace std;

int main()

{

//ОСНОВНАЯ ПРОГРАММА

setlocale(LC\_ALL, "rus");

ifstream file("D:\\file31.txt");//открытие файла

if (!file.is\_open()) cout << "Файла не существует"<<endl<<endl;

else if (file.peek() == EOF) cout << "Файл пуст"<<endl<<endl;

else

{

HANDLE hMutex;

STARTUPINFO si;//основной

PROCESS\_INFORMATION pi;//дочерний

hMutex = CreateMutex(NULL, FALSE, L"DemoMutex");// создаем мьютекс

if (hMutex == NULL)

{

cout << "Мьютекс не создан. Нажмите любую клавишу чтобы выйти" << endl;

cin.get();

return GetLastError();

}

ZeroMemory(&si, sizeof(STARTUPINFO));

si.cb = sizeof(STARTUPINFO);

// создаем новый консольный процесс

if (!CreateProcess(L"D:\\ConsoleProcess.exe", NULL, NULL, NULL, FALSE,

NULL, NULL, NULL, &si, &pi))

{

cout << "Процесс не создан. Нажмите любую клавишу чтобы выйти" << endl;

cin.get();

return GetLastError();

}

string str;

while (!file.eof()) //выполняется пока файл не закрылся

{

WaitForSingleObject(hMutex, INFINITE);// захватываем мьютекс

for (int i = 0; i < 3; i++)

{

if (file.eof()) break;

getline(file, str);//получение строки из файла

cout << "\tРодитель. процесс: " << str << endl;

Sleep(100);

}

cout << endl;

ReleaseMutex(hMutex);// освобождаем мьютекс

}

CloseHandle(hMutex);// закрываем дескриптор мьютекса

WaitForSingleObject(pi.hProcess, INFINITE);// ждем пока дочерний процесс закончит работу

CloseHandle(pi.hThread);// закрываем дескрипторы дочернего процесса в текущем процессе

CloseHandle(pi.hProcess);

}

file.close();

system("pause");

return 0;

}

**Дочерняя программа**

#include "stdafx.h"

#include <windows.h>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

//ДОЧЕРНЯЯ ПРОГРАММА

setlocale(LC\_ALL, "rus");

ifstream file("D:\\file31.txt");//открытие файла

if (!file.is\_open()) cout << "Файла не существует" << endl << endl;

else if (file.peek() == EOF) cout << "Файл пуст" << endl << endl;

else

{

HANDLE hMutex;

// открываем мьютекс

hMutex = OpenMutex(SYNCHRONIZE, FALSE, L"DemoMutex");

if (hMutex == NULL)

{

cout << "Не открыть мьютекс. Нажмите любую клавишу чтобы выйти" << endl;

cin.get();

return GetLastError();

}

string str;

while (!file.eof()) {

WaitForSingleObject(hMutex, INFINITE);// захватываем мьютекс

for (int i = 0; i < 5; i++)

{

if (file.eof()) break;

getline(file, str);//получениестрокиизфайла

cout << "Дочерний процесс: " << str << endl;

Sleep(5);

}

cout << endl;

ReleaseMutex(hMutex);// освобождаем мьютекс

}

CloseHandle(hMutex);// закрываем дескриптор объекта

}file.close();return 0;}

**План тестирования:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Классы эквивалентности** | **Тестовый набор** | **Ожидаемые результаты** |
| 1 | Класс допустимых значений  Файл содержит строки | имя файла – «file31.txt»  путь к дочернему процессу – «D:\\ConsoleProcess.exe» | При допустимом значении, программа распечатывает строки текста из файла так, чтобы вывод родительского и дочернего процесса был синхронизован. (рис. 1) |
| 2 | Класс не допустимых значений  Файл пуст | имя файла – «empty.txt» | «Файл пуст»  Программа проверяет, не пуст ли файл и если да, то выводит соответствующее сообщение.  (рис. 2) |
| 3 | Класс не допустимых значений  Файл не существует | имя файла – «iexistISWEAR.txt» | «Файл не существует»  Программа проверяет, существует ли файл и если нет, то выводит соответствующее сообщение.  (рис. 3) |
| 4 | Класс не допустимых значений  Процесс не существует | путь к дочернему файлу – «D:\\WRONGADRESS.exe» | «Процесс не создан»  Программа проверяет, создался ли процесс и если нет, то выводит соответствующее сообщение.  (рис. 4) |

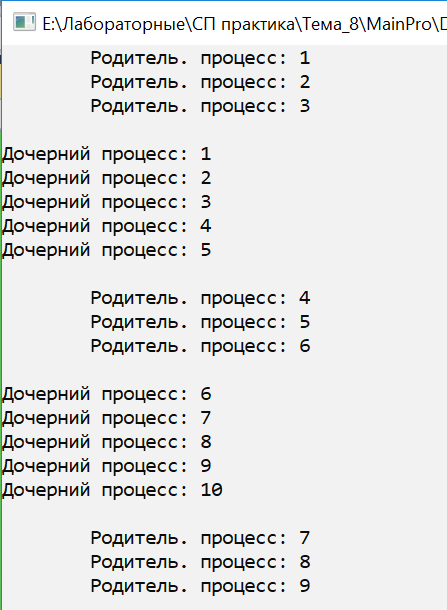


Рисунок 1

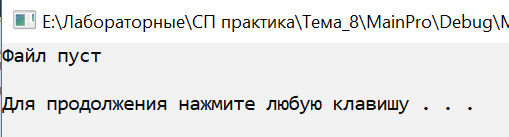


Рисунок 2

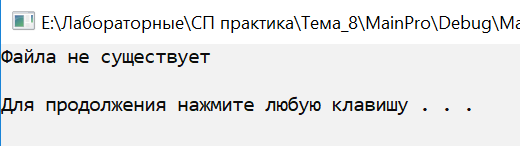


Рисунок 3

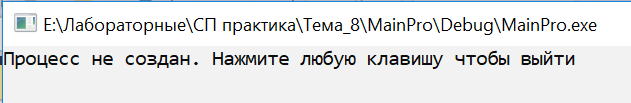


Рисунок 4