**Министерство науки и высшего образования**

**Российской Федерации**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**институт фундаментальных наук**

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине Языки программирования

Тема: **“Структуры”**

Группа: ФИТ-194

Студент: Бауэр Рудольф Евгеньевич

Преподаватель: Русакова Нина Александровна

Кемерово, 2020

Ссылка на git-репозиторий: https://github.com/Exsolg/Programming-language/tree/Lab1-Sem3

**Задания:**

1. Необходимо описать объект «Дата» с помощью структуры.

2. Для структуры предусмотреть методы: вывод информации, получение следующей даты, получение предыдущей даты.

3. В главной программе необходимо создать массив из n объектов. Инициализировать массив данными, считанными из файла.

Вывести информацию о всех объектах на экран в формате

<номер> <дата> <следующая дата> <предыдущая дата>

**Задания по вариантам:**

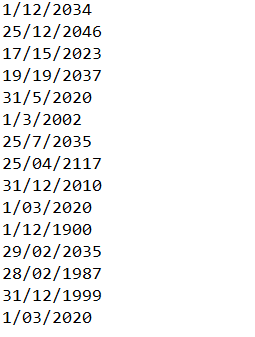
Вариант 2:

Выбрать из массива исходных дат те, для которых следующие попадут на другой год. Вывести информацию о таких датах на экран в формате:

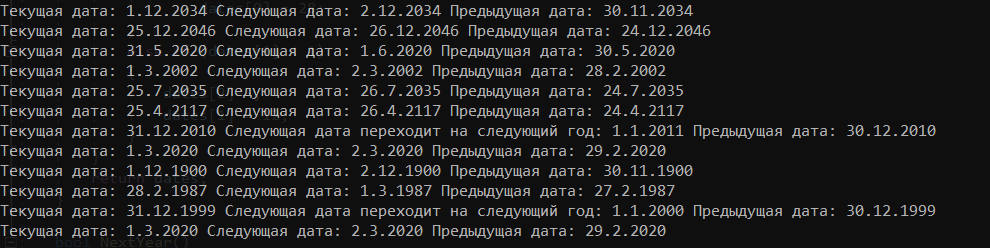
<номер> <дата> <следующая дата>

**Тесты:**

Вот даты из файла:



При запуске программы нам вывелось 12 дат из 15



Даты 17.15.2023, 19.19.2037 и 29.02.2035 нам не вывелись, так как они не прошли проверку.

Дата 28.02.1987 получила следующую дату 01.03.1987, так как 87-ой год не был високосным

Дата 01.03.2002 получила предыдущую дату 28.02.2002, так как 2002-ой год не был високосным. Но дата 01.03.2020 получила предыдущую дату 29.02.2020, потому что 2020 год является високосным.

И в задании по вариантам, те даты для которых следующие выпадают на следующий год - 31.12.2010 и 31.12.1999. Как видим, в программе нам написало, что они переходят на следующий год.

**Код программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <tuple>

#include <string>

#include <fstream>

using namespace std;

class DATE

{

public:

int day;

int month;

int year;

bool leap;

void DateOut()

{

cout << day << "." << month << "." << year << endl;

}

void leapcheck()

{

if ((year % 400 == 0 || year % 4 == 0) && year % 100 != 0) leap = true;

else leap = false;

}

int\* NextDate()

{

int\* dates = new int[3];

leapcheck();

dates[0] = day + 1;

dates[1] = month;

dates[2] = year;

for (int i = 0; i < 12; i++)

{

if ((dates[1] == 1 || dates[1] == 3 || dates[1] == 5 || dates[1] == 7 || dates[1] == 8 || dates[1] == 10 || dates[1] == 12) && dates[0] > 31)

{

dates[1]++;

dates[0] = 1;

}

else if ((dates[1] == 4 || dates[1] == 6 || dates[1] == 7 || dates[1] == 11) && dates[0] > 30)

{

dates[1]++;

dates[0] = 1;

}

else if (dates[1] == 2 && dates[0] > (28 + leap))

{

dates[1]++;

dates[0] = 1;

}

else if (dates[1] > 12)

{

dates[2]++;

dates[1] = 1;

}

}

return dates;

}

int\* PreDate()

{

int\* dates = new int[3];

leapcheck();

dates[0] = day - 1;

dates[1] = month;

dates[2] = year;

for (int i = 0; i < 12; i++)

{

if ((dates[1] == 2 || dates[1] == 4 || dates[1] == 6 || dates[1] == 8 || dates[1] == 9 || dates[1] == 11 || dates[1] == 1) && dates[0] < 1)

{

dates[1]--;

dates[0] = 31;

}

else if ((dates[1] == 5 || dates[1] == 7 || dates[1] == 8 || dates[1] == 12) && dates[0] < 1)

{

dates[1]--;

dates[0] = 30;

}

else if (dates[1] == 3 && dates[0] < 1)

{

dates[1]--;

if(leap == true)

dates[0] = 29;

else

dates[0] = 28;

}

else if (dates[1] < 1)

{

dates[2]--;

dates[1] = 12;

}

}

return dates;

}

bool NextYear()

{

bool NY;

int\* dates = NextDate();

if (dates[0] == 1 && dates[1] == 1)

NY = true;

else NY = false;

return NY;

}

};

tuple <DATE\*, int> DateFin();

void CheckDate(int\*\* arr, int cnt);

int main()

{

setlocale(LC\_ALL, "ru");

DATE\* date = get<0>(DateFin());

int\* NDate;

int\* PDate;

bool NY;

for (int i = 0; i < get<1>(DateFin()); i++)

{

NDate = date[i].NextDate();

PDate = date[i].PreDate();

NY = date[i].NextYear();

if (NY == true)

{

cout << "Текущая дата: " << date[i].day << "." << date[i].month << "." << date[i].year << " Следующая дата переходит на следующий год: ";

cout << NDate[0] << "." << NDate[1] << "." << NDate[2] << " Предыдущая дата: ";

cout << PDate[0] << "." << PDate[1] << "." << PDate[2] << endl;

}

else

{

cout << "Текущая дата: " << date[i].day << "." << date[i].month << "." << date[i].year << " Следующая дата: ";

cout << NDate[0] << "." << NDate[1] << "." << NDate[2] << " Предыдущая дата: ";

cout << PDate[0] << "." << PDate[1] << "." << PDate[2] << endl;

}

}

return 0;

}

tuple <DATE\*, int> DateFin()

{

ifstream date("RDate.txt");

char date\_inf[255];

char\* buf;

int cnt = 0, i = 0, cnt2 = 0, c = 0;

while (date.getline(date\_inf, 255, '\n')) cnt++;

int\*\* inf = new int\* [cnt];

for (int i = 0; i < cnt; i++) inf[i] = new int[3];

date.close();

date.open("RDate.txt");

while (date.getline(date\_inf, 255, '\n')) //читаем информацию с файла по разделтелям

{

buf = strtok(date\_inf, "/");

inf[i][0] = stoi(buf);

buf = strtok(NULL, "/");

inf[i][1] = stoi(buf);

buf = strtok(NULL, "/");

inf[i][2] = stoi(buf);

i++;

}

date.close();

CheckDate(inf, cnt);

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

if (inf[i][0] != 0) cnt2++;

DATE\* date\_info = new DATE[cnt2];

i = 0;

while (i < cnt && c < cnt2)

{

if (inf[i][0] != 0)

{

date\_info[c].day = inf[i][0];

date\_info[c].month = inf[i][1];

date\_info[c].year = inf[i][2];

c++; i++;

}

else i++;

}

return make\_tuple(date\_info, cnt2);

}

void CheckDate(int\*\* arr, int cnt)

{

int months[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

bool leap = false;

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

{

if (((arr[i][2] % 4 == 0) && (arr[i][2] % 100 != 0)) || (arr[i][2] % 400 == 0)) leap = true;

else leap = false;

if (arr[i][2] < 1) arr[i][0] = 0;

else if (arr[i][1] < 1 || arr[i][1] > 12) arr[i][0] = 0;

else if ((arr[i][1] == months[0] || arr[i][1] == months[2] || arr[i][1] == months[4] || arr[i][1] == months[6] || arr[i][1] == months[7] || arr[i][1] == months[9] || arr[i][1] == months[11]) && arr[i][0] > 31) arr[i][0] = 0;

else if ((arr[i][1] == months[3] || arr[i][1] == months[5] || arr[i][1] == months[8] || arr[i][1] == months[10]) && arr[i][0] > 30) arr[i][0] = 0;

else if (arr[i][1] == 2 && arr[i][0] > (28 + leap)) arr[i][0] = 0;

}

}