Федеральное агенство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

"Сибирский государственный университет телекоммуникаций и информатики"

Курсовая работа

по дисциплине "Программирование"

ВАРИАНТ \*

Выполнил: студент группы \*

\*

Проверил: ассистент кафедры ПМиК

Дьячкова И.С.

Новосибирск 2020

**Содержание:**

1. **Введение**
2. **Описание алгоритма**
3. **Код программы на C++**
4. **Результат работы программы**

**1. Введение**

**Задача:**

Выполнить Курсовую работу по дисциплине "Программирование" за 2 семестр

**Описание:**

Согласно заданию для моего варианта необходимо: «Написать программу, выводящую на экран и в файл календарь на заданный год».

**2. Описание алгоритма**

**Функции**

void calendar\_init(); - обнуляет массивы месяцов в структуру календаря

int MonthFill(int mass[n], int shift, int max\_days); - заполняет месяц, записывая числа с учётом сдвига по неделе

void YearFill(int shift, int year); - вызывает заполнение месяца для всех массивов в структуре

void month\_name\_print(int mn); -выводит определённое название месяца в консоль

void hole\_year\_print(int year); - выводит календарь определённого года в консоль

int menu(int current\_year, int current\_shift, int \*need\_year, int \*need\_shift); - выводит меню

void new\_shift(int \*shift,int i); - рассчитывает сдвиг по неделе для следующего месяца в году

int calculate\_shift(int current\_year, int need\_year, int shift); - рассчитывает сдвиг по неделе для нового года

int calendar\_save(char filename[256], int year); - сохраняет показываемый год в текстовый документ

**Переменные**

Структуры, используемые для вычисления текущего года

time\_t t;

tm \*tk;

Максимальное количество ячеек в массиве месяца

const int n = 42;

Структура календаря, содержащая массивы месяцов

struct calendar\_struc {

int January[n];

int February[n];

int March[n];

int April[n];

int May[n];

int June[n];

int July[n];

int August[n];

int September[n];

int October[n];

int November[n];

int December[n];

} Calendar;

3**. Код программы на C++**

##include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <iostream>

using namespace std;

time\_t t;

tm \*tk;

const int n = 42;

struct calendar\_struc {

int January[n];

int February[n];

int March[n];

int April[n];

int May[n];

int June[n];

int July[n];

int August[n];

int September[n];

int October[n];

int November[n];

int December[n];

} Calendar;

void calendar\_init();

int MonthFill(int mass[n], int shift, int max\_days);

void YearFill(int shift, int year);

void month\_name\_print(int mn);

void hole\_year\_print(int year);

int menu(int current\_year, int current\_shift, int \*need\_year, int \*need\_shift);

void new\_shift(int \*shift,int i);

int calculate\_shift(int current\_year, int need\_year, int shift);

int calendar\_save(char filename[256], int year);

void calendar\_init(){

for(int i=0;i<n;i++){

Calendar.January[i]=0;

Calendar.February[i]=0;

Calendar.March[i]=0;

Calendar.April[i]=0;

Calendar.May[i]=0;

Calendar.June[i]=0;

Calendar.July[i]=0;

Calendar.August[i]=0;

Calendar.September[i]=0;

Calendar.October[i]=0;

Calendar.November[i]=0;

Calendar.December[i]=0;

}

return;

}

int MonthFill(int mass[n], int shift, int max\_days){

int i=0,v=0;

for(i=0,v=0;i<n;i++,v++){

mass[i+shift]=i+1;

if(v==max\_days-1)

break;

}

return i+shift;

}

void YearFill(int shift, int year){

int i=0;

i = MonthFill(Calendar.January,shift,31);

new\_shift(&shift,i);

if(year % 4 !=0)

i = MonthFill(Calendar.February,shift,28);

else

i = MonthFill(Calendar.February,shift,29);

new\_shift(&shift,i);

i = MonthFill(Calendar.March,shift,31);

new\_shift(&shift,i);

i = MonthFill(Calendar.April,shift,30);

new\_shift(&shift,i);

i = MonthFill(Calendar.May,shift,31);

new\_shift(&shift,i);

i = MonthFill(Calendar.June,shift,30);

new\_shift(&shift,i);

i = MonthFill(Calendar.July,shift,31);

new\_shift(&shift,i);

i = MonthFill(Calendar.August,shift,31);

new\_shift(&shift,i);

i = MonthFill(Calendar.September,shift,30);

new\_shift(&shift,i);

i = MonthFill(Calendar.October,shift,31);

new\_shift(&shift,i);

i = MonthFill(Calendar.November,shift,30);

new\_shift(&shift,i);

i = MonthFill(Calendar.December,shift,31);

new\_shift(&shift,i);

}

void month\_name\_print(int mn){

switch(mn){

case 1:

cout << "Январь";

break;

case 2:

cout << "Февраль";

break;

case 3:

cout << "Март";

break;

case 4:

cout << "Апрель";

break;

case 5:

cout << "Май";

break;

case 6:

cout << "Июнь";

break;

case 7:

cout << "Июль";

break;

case 8:

cout << "Август";

break;

case 9:

cout << "Сентябрь";

break;

case 10:

cout << "Октябрь";

break;

case 11:

cout << "Ноябрь";

break;

case 12:

cout << "Декабрь";

break;

}

return;

}

void hole\_year\_print(int year){

cout << "\t\t\t\t\t " << year << " год\n";

cout << "\t";

month\_name\_print(1);

cout << "\t\t\t";

month\_name\_print(2);

cout << "\t\t\t ";

month\_name\_print(3);

cout << "\t\t\t ";

month\_name\_print(4);

cout << "\n ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС\n";

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.January[i\*7+v]!=0)

printf("%02d",Calendar.January[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.February[i\*7+v]!=0)

printf("%02d",Calendar.February[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.March[i\*7+v]!=0)

printf("%02d",Calendar.March[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.April[i\*7+v]!=0)

printf("%02d",Calendar.April[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << endl;

}

cout << endl;

cout << "\t";

month\_name\_print(5);

cout << "\t\t\t ";

month\_name\_print(6);

cout << "\t\t\t ";

month\_name\_print(7);

cout << "\t\t\t ";

month\_name\_print(8);

cout << "\n ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС\n";

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.May[i\*7+v]!=0)

printf("%02d",Calendar.May[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.June[i\*7+v]!=0)

printf("%02d",Calendar.June[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.July[i\*7+v]!=0)

printf("%02d",Calendar.July[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.August[i\*7+v]!=0)

printf("%02d",Calendar.August[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << endl;

}

cout << endl;

cout << "\t";

month\_name\_print(9);

cout << "\t\t";

month\_name\_print(10);

cout << "\t\t\t ";

month\_name\_print(11);

cout << "\t\t\t ";

month\_name\_print(12);

cout << "\n ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС";

cout << "\t ПН ВТ СР ЧТ ПТ СР ВС\n";

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.September[i\*7+v]!=0)

printf("%02d",Calendar.September[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.October[i\*7+v]!=0)

printf("%02d",Calendar.October[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.November[i\*7+v]!=0)

printf("%02d",Calendar.November [i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.December[i\*7+v]!=0)

printf("%02d",Calendar.December[i\*7+v]);

else

cout << " ";

cout << " ";

}

cout << endl;

}

cout << endl;

return;

}

int menu(int current\_year, int current\_shift, int \*need\_year, int \*need\_shift){

int shift;

char enter;

char input[256];

char filename[256] = "CalendarSave.txt";

cout << endl << " 1 - показать текущий год";

cout << endl << " 2 - выбрать год";

cout << endl << " 3 - сохранить текущий календарь";

cout << endl << " 4 - выйти\n ";

cin >> enter;

switch(enter){

case '1':

\*need\_year = current\_year;

\*need\_shift = current\_shift;

system("CLS");

calendar\_init();

YearFill(current\_shift,current\_year);

break;

case '2':

cout << endl << " Введите нужный год: ";

cin >> input;

\*need\_year = atoi(input);

system("CLS");

calendar\_init();

shift = calculate\_shift(current\_year, \*need\_year, current\_shift);

YearFill(shift,\*need\_year);

break;

case '3':

return calendar\_save(filename, \*need\_year);

break;

case '4':

return -1;

default:

cout << "\n Неправильный ввод";

return 0;

}

return 0;

}

void new\_shift(int \*shift,int i){

if (i<34)

\*shift = 7-(34-i);

else

\*shift = 7-(41-i);

return;

}

int calculate\_shift(int current\_year, int need\_year, int shift){

int counter;

if(current\_year>need\_year){

for(int i = current\_year; i>need\_year;i--){

if((i % 4) == 1){

shift = shift - 2;

}

else{

shift-- ;

}

if(shift<0){

shift = shift + 7;

}

}

}

else{

for(int i = current\_year; i<need\_year;i++){

if((i % 4) == 1){

shift = shift+2;

}

else{

shift++;

}

if(shift>6){

shift = shift -6;

}

}

}

return shift;

}

int calendar\_save(char filename[256], int year){

FILE \*f = fopen(filename, "wb");

if(f==NULL)

{

perror("Error occured while opening file");

return -1;

}

fprintf(f,"\t\t\t\t\t %d год\n",year);

fprintf(f,"\tЯнварь\t\t\tФевраль\t\t\t Март\t\t\t Апрель\n");

fprintf(f," ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\n");

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.January[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.January[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.February[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.February[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.March[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.March[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.April[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.April[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f,"\n");

}

fprintf(f,"\n");

fprintf(f,"\tМай\t\t\t Июнь\t\t\t Июль\t\t\t Август\n");

fprintf(f," ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\n");

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.May[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.May[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.June[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.June[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.July[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.July[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.August[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.August[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f,"\n");

}

fprintf(f,"\n");

fprintf(f,"\tСентябрь\t\tОктябрь\t\t\t Ноябрь\t\t\t Декабрь\n");

fprintf(f," ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\t ПН ВТ СР ЧТ ПТ СР ВС\n");

for(int i=0;i<6;i++){

cout << " ";

for (int v=0; v<7;v++){

if(Calendar.September[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.September[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.October[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.October[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.November[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.November[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f," ");

for (int v=0; v<7;v++){

if(Calendar.December[i\*7+v]!=0)

fprintf(f,"%02d",Calendar.December[i\*7+v]);

else

fprintf(f," ");

fprintf(f," ");

}

fprintf(f,"\n");

}

fprintf(f,"\n");

fclose(f);

return 0;

}

int main(){

setlocale(LC\_ALL,"RUS");

srand(time(NULL));

time(&t);

tk=localtime(&t);

int current\_year =1900+tk->tm\_year;

int current\_shift = 2;

int need\_year = current\_year;

int need\_shift = current\_shift;

calendar\_init();

YearFill(current\_shift,current\_year);

while(true){

system("CLS");

hole\_year\_print(need\_year);

int r = menu(current\_year, current\_shift, &need\_year, &need\_shift);

if(r == -1)

break;

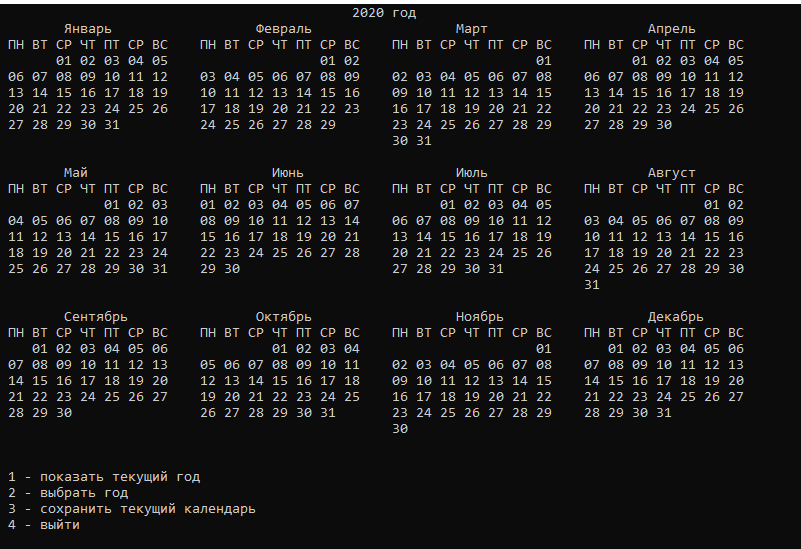
}

system("pause");

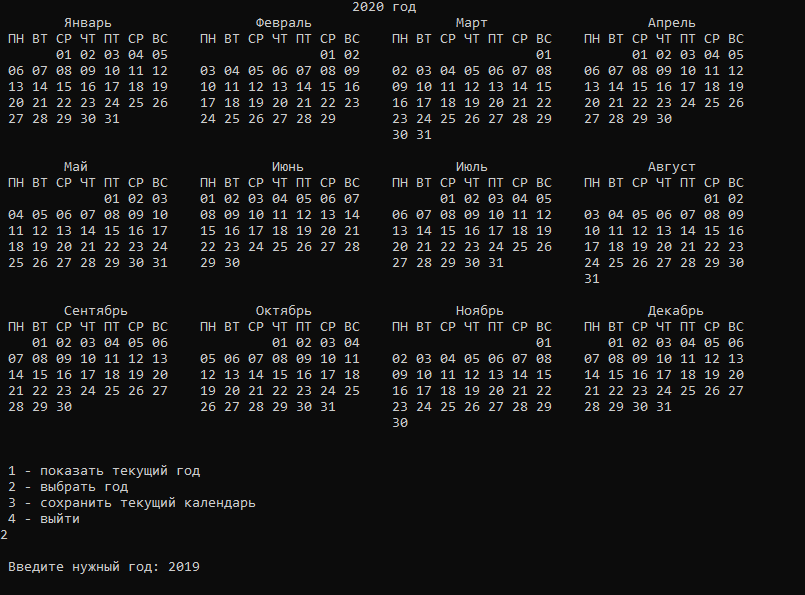
return 0;

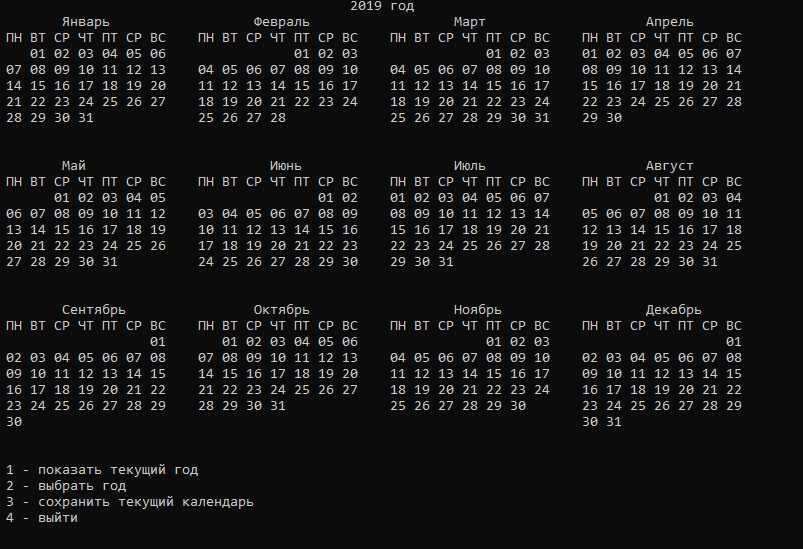
}**4. Результат работы программы**

1.начало

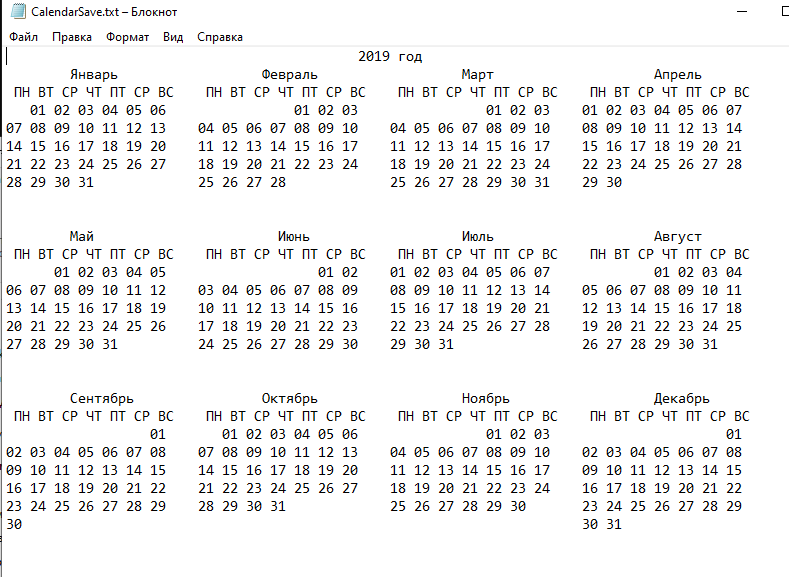


2. Выбор года





3. Сохранение выбранного года



4.Возврат к текущему году

