|  |
| --- |
| Министерство Образования Республики Беларусь Белорусский Государственный Университет  Информатики и Радиоэлектроники  Кафедра ЭВМ Отчёт по учебной практике   Выполнил : Проверил :  студент гр.050505 Калабухов Е.В.  Свентицкий С. М.    Минск  2001 г. |

СОДЕРЖАНИЕ

1. Название программы.
2. Поставленная задача.
3. Функциональное предназначение программы.
4. Описание управления, интерфейса, основных переменных и функций.
5. Блок-схема по ЕСПД 19.003-80.
6. Исходный код программы.
7. Список используемой литературы

1. Название программы2. Поставленная задача

#### Electronic Base (Электронная база данных) v 1.666

Написать базу данных (БД) на языке С++ позволяющую хранить информацию о студентах (Ф.И.О., номер группы, средний балл, уникальный код записи (порядковый №)). База данных должна содержать основные функции : добавление записи, удаление записи, редактирование записи, поиск и сортировку по заданным параметрам.

3.Функциональное предназначение программы

Данная программа представляет собой базу данных позволяющую хранить информацию о студентах (Фамилию, Имя, Отчество, № группы, средний балл). Программа позволяет использовать следующие возможности : создание новой БД, либо открытий уже созданной, добавление новых данных в файл БД, удаление записей, редактирование имеющихся записей, поиск существующих записей по их уникальному коду, по фамилии, по имени, также предусмотрена сортировка записей в файле по Ф.И.О, по уникальному коду, по номеру группы, по среднему баллу (причем в этом случае записи располагаются от максимального среднего балла вниз, то есть по убыванию). Каждой новой записи присваивается свой индивидуальный номер (код), для упрощения ввода этот код присваивается автоматически : функций **find\_max\_num(char \*)** находит максимальный номер записи в файле и последующий код присваивается введенной записи .Также предусмотрена проверка вводимых данных на идентичность , при вводе идентичных данных пользователю будет выдано предупреждение и дана возможность ввести новые данные . Программа **eBase v1.666** использует очень удобный и понятный интерфейс, организована система меню и подменю, внизу окна специально создана горячая подсказка. Так же предусмотрен так называемый скроллинг (прокрутка) выводимой информации клавишами вверх-вниз,

4. Описание управления, интерфейса, основных переменных и функций.

**Все данные вводятся с клавиатуры.**

**Используемые клавиши:**

**Enter** (для выбора или подтверждения), **Esc** (для отмены, выхода в предыдущее меню), **стрелки вправо, влево** (для перемещения по меню), **вверх, вниз** (для скроллинга данных).

**Основные переменные:**

структура записей :

**typedef struct {**

**int num;** - уникальный № (код) записи

**char lastName[16];** - фамилия студента

**char firstName[11];** - имя студента

**char midName[16];** - отчество студента

**char groupNumber[6];** - № группы студента

**float mark;** - средний балл учащегося

**} data;**

**fn** - имя рабочего файла;

**Вызываемые функции:**

**void interface(char \*)** - функция основного интерфейса программы. В качестве входного параметра принимает имя рабочего файла **fn** которое передается функциями **void create()** или **void open()**. В этой функции реализовано весьма удобное меню, через которое и происходит вызов последующий функция добавления, сортировки, удаления, поиска и редактирования;

**void add(char \*)** - функция добавление записи в БД. Входной параметр **fn** имя файла. Использует функцию **int find\_max\_num(char \*)** для присваивания вводимой записи свой уникальный номер.

**void del(char \*)** - функция удаление записи из БД. Входной параметр **fn** имя файла. Использует временный файл **temp.$$$**, в конце выполнения функции временный файл переименовывается в файл под именем переменной **fn**;

**void edit(char \*)** - функция редактирования существующей записи. Входной параметр **fn** имя файла. Позволяет изменять № (код) записи, а также любую информацию записи;

**void view(char \*, int )** - функция вывода данных на экран. Входной параметр **fn** имя файла;

**void find(char \*)** - функция поиска записи по (ее уникальному №,по Фамилии, по Имени). Входной параметр **fn** имя файла;

**void sort(char \*)** - функция сортировки записей БД по (№ записей, Ф.И.О, среднему баллу учащихся, № группы студента). Входной параметр **fn** имя файла. Использует “пузырьковый” метод сортировки данных в файле.

**int find\_max\_num(char \*)** - функция поиска максимального № записи в файле БД. Входной параметр **fn** имя файла. Возвращает значение № максимальной записи в файле … для функции **void add(char \*)**;

**int get\_number(char \*)** - возвращает количество записей в файле. Входной параметр **fn** имя файла;

**void create()** - создание файла БД. Идет запрос на имя файла , после этого к имени файла автоматически добавляется расширение “.dat” имя этого файла считается рабочим для всей программы. При удачном **создании** файла его имя передается в функцию **void interface(char \*)** и т.д.;

**void open()** - открытие файла БД. Идет запрос на имя файла , после этого к имени файла автоматически добавляется расширение “.dat” имя этого файла считается рабочим для всей программы. При удачном **открытии** файла его имя передается в функцию **void interface(char \*)** и т.д.;

**int check(char \*, data)** – функция проверки при вводе информации на ее идентичность. Вызывается функциями добавления записи **void add(char \*)** и редактирования записи **void edit(char \*)**. В качестве первого входного значения используется имя рабочего файла **fn** , в качестве второго структура данных типа **data** (описанная в разделе основные переменные). В качестве возвращаемого параметра используется переменная **pr**,которая может принимать значения 0 или 1, для определения идентичности данных;

# **Нестандартные и системные средства, вроде бы, не используются**

Все, кроме двух, аварийные ситуации, вроде бы, ликвидированы:

1) Если диск заполнен или защищён от записи невозможно записать информацию в файл;

2) Если невозможно открыть файл программа выдает сообщение об этом и закроется

5.Блок-схема по ЕСПД 19.003-80.

void main()

### Лист 01

Конец

Меню

выбора

open()

Начало

#### void open() void interface(char \*)

create()

fopen

Конец

strcat(fn,”dat”)

нет

interface(fn)

да

fn

Начало

#### алгоритм сортировки (по № записи)

01

view(fn,num)

Начало

sort(fn)

find(fn)

edit(fn)

Меню

выбора

add(fn)

del(fn)

**fp** – указатель на файл; **s1,s2** – структуры данных типа data;

**oldfs,newfs** – флаги (признаки),

#### 6. Исходный код программы

нет

да

нет

s1=s2

да

да

newfs=1;

oldfs=1;

newfs=0;

Конец

oldfs=1;

newfs=0;

Начало

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <io.h>

#include <iostream.h>

#include <string.h>

#include <bios.h>

#define ENTER 7181

#define ESC 283

#define LEFT 19200

#define RIGHT 19712

#define UP 18432

#define DOWN 20480

typedef struct {

int num;

char lastName[16];

char firstName[11];

char midName[16];

char groupNumber[6];

float mark;

} data;

void interface(char \*);

void add(char \*);

void del(char \*);

void edit(char \*);

void view(char \*, int );

void find(char \*);

void sort(char \*);

int find\_max\_num(char \*);

int get\_number(char \*);

void create();

void open();

int check(char \*, data);

void main()

{

int key=0,counter=0;

int color[]={0,4,4};

while (1) {

textbackground(1); textcolor(WHITE);

clrscr(); \_setcursortype(\_NOCURSOR); textcolor(10);

cprintf("::::::::::::::::::");

textcolor(15);

cprintf(" eBase v1.666 Copyright (C) Serj Sventitski ");

textcolor(10);

cprintf("::::::::::::::::::");

switch(key) {

case LEFT: if (counter>0) {

color[counter]=4;

color[--counter]=0;

}

break;

case RIGHT: if (counter<2) {

color[counter]=4;

color[++counter]=0;

}

break;

}

textbackground(4); cprintf(" ");

textcolor(WHITE);

textbackground(color[0]); cprintf(" Создать ");

textbackground(color[1]); cprintf(" Открыть ");

textbackground(color[2]); cprintf(" Выход ");

textbackground(4); cprintf(" ");

textbackground(1);

key=bioskey(0);

if (key == ENTER) {

switch(counter) {

case 0:

create();break;

case 1:

open();break;

case 2:

exit(10);

}

}

}

}

void interface(char \*fn)

{

FILE \*fp;

data st;

int num=1,key=0,counter=0,color[]={0,4,4,4,4,4};

clrscr();

while(1) {

window(1,1,80,3);

textbackground(1); textcolor(WHITE);

\_setcursortype(\_NOCURSOR); textcolor(10);

cprintf("::::::::::::::::::");

textcolor(15);

cprintf(" eBase v1.666 Copyright (C) Serj Sventitski ");

textcolor(10);

cprintf("::::::::::::::::::");

switch(key)

{

case LEFT: if (counter>0) {

color[counter]=4;

color[--counter]=0;

}

break;

case RIGHT: if (counter<5) {

color[counter]=4;

color[++counter]=0;

}

break;

}

textbackground(4);cprintf(" ");

textcolor(WHITE);

textbackground(color[0]);cprintf(" Добавить ");

textbackground(color[1]);cprintf(" Удалить ");

textbackground(color[2]);cprintf(" Редактировать ");

textbackground(color[3]);cprintf(" Поиск ");

textbackground(color[4]);cprintf(" Сортировка ");

textbackground(color[5]);cprintf(" Назад ");

textbackground(4);cprintf(" ");

textbackground(1);

view(fn,num);

key=bioskey(0);

switch(key)

{

case UP: num--;if (num==0)

num++;

view(fn,num);

break;

case DOWN: num++;if (num+16>=get\_number(fn))

num--;

view(fn,num);

break;

}

if (key==283)

return;

if (key == ENTER)

{

switch(counter) {

case 0: add(fn); clrscr(); break;

case 1: del(fn); break;

case 2: edit(fn); clrscr(); break;

case 3: find(fn); break;

case 4: sort(fn); break;

case 5: return;

}

}

}

}

void add(char \*fn)

{

FILE \*fp;

data st;

int pr;

while (1) {

st.num=find\_max\_num(fn)+1;

clrscr();

\_setcursortype(\_SOLIDCURSOR);

gotoxy(20,6); cout<<"Фамилия : "; cin>>st.lastName;

gotoxy(20,8); cout<<"Имя : "; cin>>st.firstName;

gotoxy(20,10); cout<<"Отчество : "; cin>>st.midName;

gotoxy(20,12); cout<<"№ группы : "; cin>>st.groupNumber;

gotoxy(20,14); cout<<"Средний балл студента : ";

fflush(stdin);scanf("%f",&st.mark);

if (!check(fn, st)) {

if ((fp = fopen(fn,"ab+")) == NULL) {

printf("File could not be opened.\n");

return;

}

fwrite(&st, sizeof(data),1,fp);

fclose(fp);

return;

}

else {

gotoxy(20,18);

cout<<"Такая запись уже существует";

getch();

}

}

}

void del(char \*fn)

{

FILE \*fp,\*temp;

data st;

int pr=1;

int number;

gotoxy(4,25);

textcolor(11);

cprintf("Введите № записи для удаления : ");

textcolor(YELLOW);

gotoxy(36,25);

int key = 0;

cscanf("%d",&number);

textcolor(WHITE);

while(kbhit())

if (!(key=getch()))

key << 8;

if( key == '\r' ) {

if ((fp=fopen(fn,"rb")) == NULL) {

cout<<"error.file does not exist.";

return;

}

if ((temp=fopen("temp.$$$","wb")) == NULL) {

cout<<"error. temp file couldn't be create";

return;

}

while (fread(&st,sizeof(data),1,fp)) {

if (feof(fp))

break;

if (st.num != number)

fwrite(&st,sizeof(data),1,temp);

else

pr=0;

}

fclose(fp);

fclose(temp);

if (remove(fn) == -1) {

perror("remove");

return;

}

if (rename("temp.$$$",fn) == -1) {

perror("rename");

return;

}

if (pr) {

textcolor(11);

gotoxy(4,25);

cprintf("\r Ошибка. Нет записи с таким номером ");

getch();

textcolor(WHITE);

}

}

clrscr();

}

void edit(char \*fn)

{

FILE \*fp;

data st;

int pr=1;

int number,size=sizeof(data);

fpos\_t filepos;

gotoxy(4,25);

textcolor(11);

cprintf("Введите № записи для редактирования : ");

textcolor(YELLOW);

gotoxy(44,25);

int key = 0;

cscanf("%d",&number);

textcolor(WHITE);

while( kbhit())

if( !(key = getch()))

key << 8;

if( key == '\r' )

{

if ((fp = fopen(fn,"rb+")) == NULL)

{

printf("File could not be opened.\n");

return;

}

while(fread(&st,size,1,fp))

{

if (feof(fp))

break;

if (st.num == number)

{

fseek(fp,-1\*size,1);

fgetpos(fp,&filepos);

pr=1;

break;

}

else

pr=0;

}

if (pr)

{

while (1)

{

clrscr();

\_setcursortype(\_SOLIDCURSOR);

gotoxy(20,6);cout<<"Фамилия : ";

cin>>st.lastName;

gotoxy(20,8);cout<<"Имя : ";

cin>>st.firstName;

gotoxy(20,10);cout<<"Отчество : ";

cin>>st.midName;

gotoxy(20,12);cout<<"№ группы : ";

cin>>st.groupNumber;

gotoxy(20,14);cout<<"Средний балл студента : ";

scanf("%f",&st.mark);

if (!check(fn, st))

{

fsetpos(fp,&filepos);

fwrite(&st, sizeof(data),1,fp);

break;

}

else {

gotoxy(20,18);

cout<<"Такая запись уже существует";

getch();

}

}

}

else {

textcolor(11);

gotoxy(4,25);

cprintf("\r Ошибка... Нет записи с таким номером ");

getch();

textcolor(WHITE);

}

fclose(fp);

}

}

void view(char \*fn,int n)

{

void ramka();

FILE \*fp;

data st;

char c;

int counter=0;

if ((fp = fopen(fn,"rb")) == NULL) {

printf("File could not be opened.\n");

return;

}

ramka();

window(1,6,80,23);

fseek(fp,(n-1)\*sizeof(data),SEEK\_SET);

while (counter != 18)

{

fread(&st,sizeof(data),1,fp);

if (feof(fp))

break;

else

{

printf(" %4d %-16s %-12s %-17s %8s "

"%7.2f\n",st.num,st.lastName,st.firstName,

st.midName,st.groupNumber,st.mark);

counter++;

}

}

window(1,24,80,25);

cprintf("-----------------------------------"

"---------------------------------------------");

cprintf(" %c - вверх, %c - вниз , %c - влево,"

" %c - вправо, ESC - отмена ",0x018,0x019,0x01B,0x01A);

textcolor(10); textbackground(1); cprintf("%14s",fn);

window(1,1,80,25); textcolor(15);

fclose(fp);

}

void sort(char \*fn)

{

FILE \*fp;

data s1,s2;

int num=1,size=sizeof(data),counter=0,key=0;

int oldfs,newfs,color[]={0,4,4,4,4};

while(1)

{

\_setcursortype(\_NOCURSOR); textcolor(10);

cprintf("::::::::::::::::::::::::: М е н ю с о р т и р о в к и "

":::::::::::::::::::::::::");

switch(key)

{

case LEFT: if (counter>0) {

color[counter]=4;

color[--counter]=0;

}

break;

case RIGHT: if (counter<4) {

color[counter]=4;

color[++counter]=0;

}

break;

}

textbackground(4); cprintf(" "); textcolor(WHITE);

textbackground(color[0]);cprintf(" по № записи ");

textbackground(color[1]);cprintf(" по Ф.И.О ");

textbackground(color[2]);cprintf(" по № группы ");

textbackground(color[3]);cprintf(" по ср.балу ");

textbackground(color[4]);cprintf(" Назад ");

textbackground(4);cprintf(" "); textbackground(1);

view(fn,num);

key=bioskey(0);

switch(key)

{

case UP: num--;if (num==0)

num=1;

view(fn,num); break;

case DOWN: num++;if (num+16>=get\_number(fn))

num--;

view(fn,num); break;

}

if (key==283)

return;

if (key == ENTER)

{

if ((fp=fopen(fn,"rb+")) == NULL)

{

cout<<"error.file does not exist.";

return;

}

switch(counter) {

case 1: /\*-----------по Ф.И.О--------------------\*/

oldfs=1; newfs=0;

fread(&s1,size,1,fp);

while(oldfs) {

fread(&s2,size,1,fp);

if(feof(fp)) {

if(!newfs)

break;

oldfs=1; newfs=0;

rewind(fp);

fread(&s1,size,1,fp);

continue;

}

if (strcmp(s1.lastName,s2.lastName) > 0) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

if (strcmp(s1.lastName,s2.lastName) == 0)

if (strcmp(s1.firstName,s2.firstName) > 0) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

if (strcmp(s1.firstName,s2.firstName) == 0)

if (strcmp(s1.midName,s2.midName) > 0) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

else

s1=s2;

else

s1=s2;

}

break;

case 0: /\*------------ПО НОМЕРУ ЗАПИСИ-----------\*/

oldfs=1; newfs=0;

fread(&s1,size,1,fp);

while(oldfs) {

fread(&s2,size,1,fp);

if(feof(fp)) {

if(!newfs)

break;

oldfs=1; newfs=0;

rewind(fp);

fread(&s1,size,1,fp);

continue;

}

if (s1.num > s2.num) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

else

s1=s2;

}

break;

case 2: /\*----------ПО НОМЕРУ ГРУППЫ---------------\*/

oldfs=1; newfs=0;

fread(&s1,size,1,fp);

while(oldfs) {

fread(&s2,size,1,fp);

if(feof(fp)) {

if(!newfs)

break;

oldfs=1; newfs=0;

rewind(fp);

fread(&s1,size,1,fp);

continue;

}

if (strcmp(s1.groupNumber,s2.groupNumber) > 0) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

else

s1=s2;

}

break;

case 3: /\*---------ПО СР.БАЛУ----------------\*/

oldfs=1; newfs=0;

fread(&s1,size,1,fp);

while(oldfs) {

fread(&s2,size,1,fp);

if(feof(fp)) {

if(!newfs)

break;

oldfs=1; newfs=0;

rewind(fp);

fread(&s1,size,1,fp);

continue;

}

if (s1.mark < s2.mark) {

fseek(fp,-2\*size,1);

fwrite(&s2,size,1,fp);

fwrite(&s1,size,1,fp);

newfs=1;

}

else

s1=s2;

} break;

case 4:

fclose(fp);

return;

}

fclose(fp);

}

}

}

void find(char \*fn)

{

void ramka();

FILE\* fp;

data st;

char string[15];

int pr,prs;

int key=0,counter=0;

int color[]={0,4,4,4};

while (1) {

clrscr(); textcolor(10);

cprintf("::::::::::::::::::::::::::::: М е н ю П о и с к а "

":::::::::::::::::::::::::::::");

textcolor(WHITE);

textbackground(4);

cprintf(" ");

switch(key)

{

case LEFT: if (counter>0) {

color[counter]=4;

color[--counter]=0;

} break;

case RIGHT: if (counter<3) {

color[counter]=4;

color[++counter]=0;

} break;

}

textbackground(color[0]); cprintf(" по № записи ");

textbackground(color[1]); cprintf(" по Фамилии ");

textbackground(color[2]); cprintf(" по Имени ");

textbackground(color[3]); cprintf(" Отмена ");

textbackground(4);

cprintf(" ");

textbackground(1);

key=bioskey(0);

if (key==283)

return;

if (key == ENTER)

{

if((fp=fopen(fn,"rb"))==NULL) {

puts("File couldn't be open");

return;

}

switch(counter)

{

case 0:

int number;

pr=0;

prs=1;

cout<<"ВВЕДИТЕ НОМЕР ЗАПИСИ : ";

fflush(stdin);

scanf("%d",&number);

rewind(fp);

while(fread(&st,sizeof(data),1,fp)) {

if (feof(fp)) {

break;

}

if(st.num == number) {

if (prs) {

ramka();

gotoxy(1,6);

prs=0;

}

printf(" %4d %-16s %-12s %-17s %8s "

"%7.2f\n",st.num,st.lastName,st.firstName,

st.midName,st.groupNumber,st.mark);

pr=1;

}

}

if(!pr) {

cout<<"\nЗАПИСИ С ТАКИМ НОМЕРОМ НЕ СУЩЕСТВУЕТ";

}

getch();

break;

case 1:

prs=1;

pr=0;

cout<<"ВВЕДИТЕ ФАМИЛИЮ : ";

fflush(stdin);

cin>>string;

rewind(fp);

while(fread(&st,sizeof(data),1,fp)) {

if (feof(fp))

break;

if(!strcmp(st.lastName,string)) {

if (prs) {

ramka();

gotoxy(1,6);

prs=0;

}

printf(" %4d %-16s %-12s %-17s %8s "

"%7.2f\n",st.num,st.lastName,st.firstName,

st.midName,st.groupNumber,st.mark);

pr=1;

}

}

if(!pr) {

cout<<"\nНЕТ ДАННЫХ";

}

getch();

break;

case 2:

prs=1;

pr=0;

cout<<"ВВЕДИТЕ ИМЯ : ";

fflush(stdin);

cin>>string;

rewind(fp);

while(fread(&st,sizeof(data),1,fp)) {

if (feof(fp))

break;

if(!strcmp(st.firstName,string)) {

if (prs) {

ramka();

gotoxy(1,6);

prs=0;

}

printf(" %4d %-16s %-12s %-17s %8s "

"%7.2f\n",st.num,st.lastName,st.firstName,

st.midName,st.groupNumber,st.mark);

pr=1;

}

}

if(!pr) {

cout<<"\nНЕТ ДАННЫХ";

}

getch();

break;

case 3:

fclose(fp);

return;

}

fclose(fp);

}

}

}

void ramka()

{

window(1,3,80,6);

cprintf("------------------------------------"

"--------------------------------------------");

textcolor(14);

cprintf("%5s %-16s %-12s %-17s %-8s %12s ",

"№","Фамилия","Имя","Отчество","№ группы","Ср.балл");

textcolor(WHITE);

cprintf("------------------------------------"

"--------------------------------------------");

window(1,1,80,25);

}

int find\_max\_num(char \*fn)

{

FILE \*fp;

data st;

int max;

if ((fp = fopen(fn,"rb")) == NULL) {

printf("File could not be opened.\n");

return -1;

}

max=0;

while (fread(&st,sizeof(data),1,fp)) {

if (feof(fp))

break;

if (st.num>max)

max=st.num;

}

fclose(fp);

return max;

}

int get\_number(char \*fn)

{

FILE \*fp;

int fd,n=0;

if ((fp = fopen(fn,"rb")) == NULL)

{

printf("File could not be opened.\n");

return -1;

}

fd=fileno(fp);

n=filelength(fd)/sizeof(data);

fclose(fp);

return n;

}

void create()

{

FILE \*fp;

char \*fn;

cout<<"Введите имя новой БД : ";

fflush(stdin);

cin>>fn;

strcat(fn,".dat");

if ((fp=fopen(fn,"wb")) == NULL) {

cout<<"Ошибка...Файл не может быть создан\n";

getch();

exit(10);

}

fclose(fp);

interface(fn);

}

void open()

{

FILE \*fp;

char \*fn;

cout<<"Введите имя БД : ";

fflush(stdin);

cin>>fn;

strcat(fn,".dat");

if (!(fp=fopen(fn,"rb"))) {

cout<<"Ошибка...Файл не может быть открыт\n";

getch();

exit(10);

}

fclose(fp);

interface(fn);

}

int check(char \*fn, data st)

{

FILE \*fp;

data temp;

int pr=0;

if ((fp = fopen(fn,"rb")) == NULL) {

printf("File could not be opened.\n");

return 0;

}

while ( fread( &temp, sizeof( data ), 1, fp)) {

if (feof(fp))

break;

if (!strcmp(st.lastName,temp.lastName)

&& !strcmp(st.firstName,temp.firstName)

&& !strcmp(st.midName,temp.midName)

&& !strcmp(st.groupNumber,temp.groupNumber)

&& (st.mark == temp.mark)) {

pr=1;

break;

}

} fclose(fp);

return pr;

}

#### СПИСОК ЛИТЕРАТУРЫ

1. Харви Дейтел, Пол Дейтел “Как программировать на Си”
2. Е. М. Демидович “Основы алгоритмизации и программирования”, Минск 1999 г.
3. Керниган Б., Ритчи Д. “Язык программирования Си”, Финансы и статистика, 1992 г.