

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

Пояснительная записка

к курсовой работе по курсу
«Программирование»
на тему «Разработка программы сложной структуры методом нисходящего
программирования».

Выполнил:
Студент группы 21ВВ1
Жбанников Д.Н.

Принял:
Голотенков Н.О.


22.06.22

Пенза 2022

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ



М.А. Митрохин

« ____ » ____ 20__ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Иванникову Дмитрию Николаевичу Группа 21ВВ1

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование

Приложение: "игра крестики-нолики"

Функционал: сохранение результатов игры в файл; вывод результатов последних 15-ти игр на экран

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания "01" марта 2022г.

Дата защиты проекта "22" 05. 11

Руководитель Топоткин Л.О.

Задание получил "01" марта 2022г.

Студент ИЗЖ - Жбанников Д.Н.

Оглавление

Введение	5
Задание на курсовое проектирование	Ошибка! Закладка не определена.
1 Постановка задачи	6
2 Описание структуры программы	7
4 Описание функционала программы	8
5 Руководство пользователя.	8
6 Разработка функции на языке Assembler.	11
7 Тестирование.	11
Заключение.....	16
Приложение «Листинг программы»	18

Введение

Крестики-нолики — логическая игра между двумя противниками на квадратном поле 3 на 3 клетки или большего размера (вплоть до «бесконечного поля»). Один из игроков играет «крестиками», второй — «ноликами». В традиционной китайской игре (Гомоку) используются чёрные и белые камни.

Игроки по очереди ставят на свободные клетки поля 3×3 знаки (один всегда крестики, другой всегда нолики). Первый, выстроивший в ряд 3 своих фигуры по вертикали, горизонтали или диагонали, выигрывает. Первый ход делает игрок, ставящий крестики.

Для каждой из сторон общеизвестны алгоритмы, которые гарантируют ничью при любой игре противника, а при его ошибке позволяют выиграть. Таким образом, игра находится в состоянии «ничейной смерти».

Ниже приведены некоторые из таких стратегий. Считается, что игрок всегда соблюдает два правила, имеющие приоритет над всеми остальными:

Правило 1. Если игрок может немедленно выиграть, он это делает.

Правило 2. Если игрок не может немедленно выиграть, но его противник мог бы немедленно выиграть, сделав ход в какую-то клетку, игрок сам делает ход в эту клетку, предотвращая немедленный проигрыш.

Исходя из вышесказанного, я решил проверить свои возможности в реализации данной игры программным путем. Мой результат работы представляет собой консольное приложение, предназначенное для игры между двумя игроками, с последующей записью результатов игр в файл. В дальнейшем эти результаты можно вывести на экран.

1 Постановка задачи

Необходимо разработать игру “Крестики нолики”. Программа должна иметь возможность игры между двумя игроками (крестиком и ноликом), также должна отображать победителя.

Нужно разработать интуитивно понятный интерфейс пользователя, определить пункты названия меню. Изучить работу с файлами. Это необходимо для чёткого понимания функциональности программы.

Приложение должно отвечать следующим требованиям:

1. Многомодульность. Программа должна быть поделена на логические модули. Это упростит дальнейшее совершенствование программы, а также позволит локализовать ошибки.
2. Использование сложных типов данных: массивов, структур, объектов и т.д. Для разработки консольного мессенджера необходимо подробно изучить принципы работы с массивами и объектами для реализации алгоритмов хранения сообщений и взаимодействия с сервером.
3. Режим работы видеосистемы: текстовый/графический. Необходимо определиться с типом управления, видом интерфейса и изучить приёмы реализации графики.
4. Устройство ввода информации: клавиатура/мышь.
5. Пользовательский интерфейс должен быть построен на основе меню или панели инструментов. Необходимо реализовать пользовательское меню команд, с заставкой.
6. Операционная система: MS Windows. При разработке использовать среду MS Windows с системой программирования MS Visual Studio.
7. Язык программирования: Си и Ассемблер.

2 Описание структуры программы

Функция «main» – главная функция, в которой вызываются функции, определяющие меню, а также дальнейшее определение победителя в игре.

Схема программы изображена на рисунке

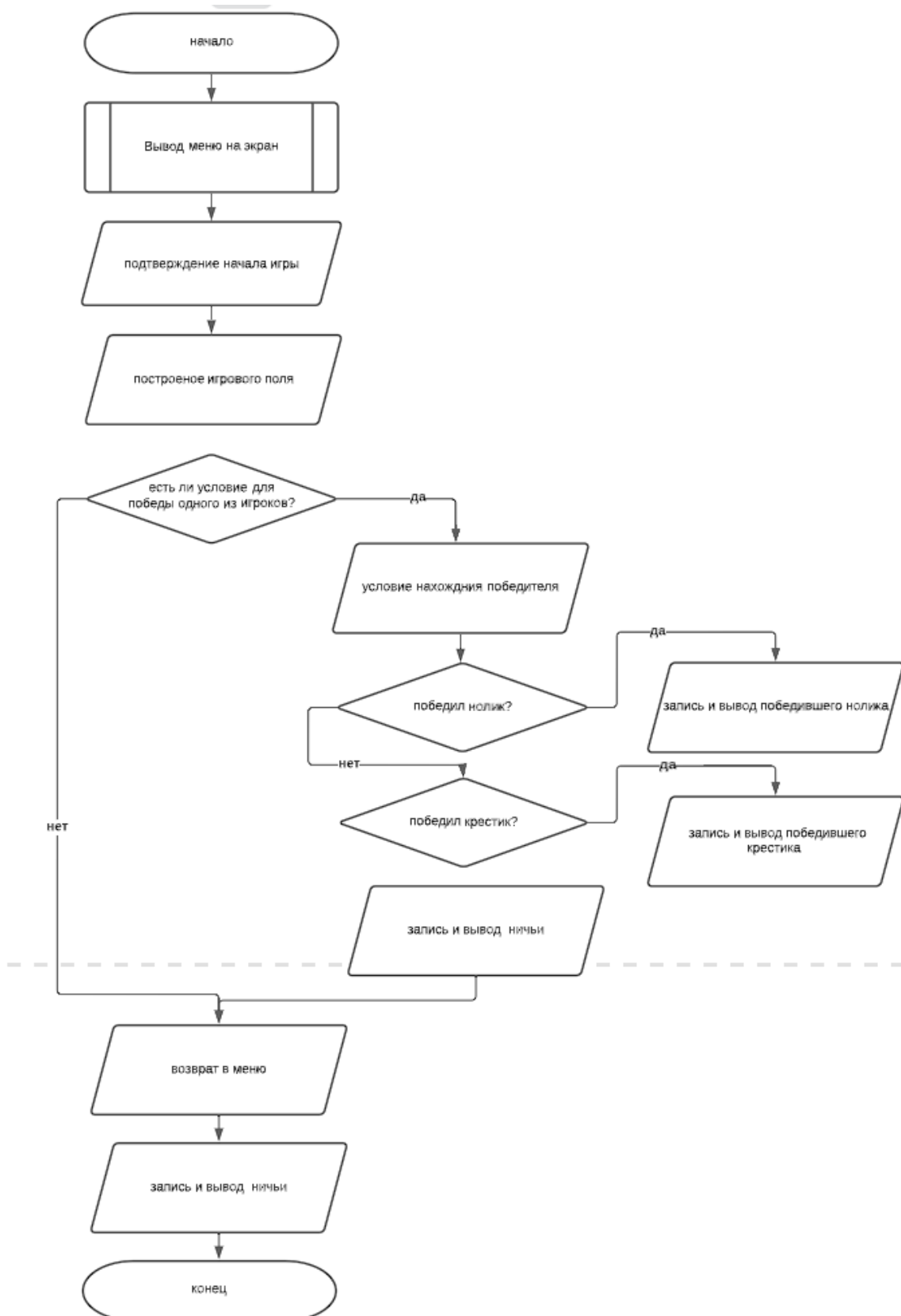
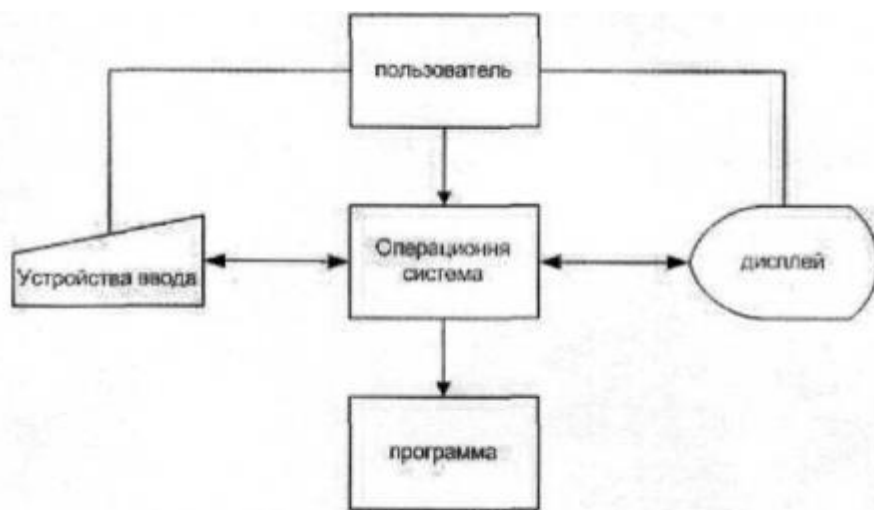


Рисунок – структура программы.
Схема данных программы



4 Описание функционала программы.

Функция «potain» – главная функция. В ней происходит запуск всех важных функций. Здесь отображается игровое поле, а также происходит проверка победителя

Далее в функции poll() пробегаем по всем строкам.

В следующих файлах идет графическое представление программы.

Proverka() создана для нахождения выигрышной комбинации, где выполняется проверка всех возможных комбинаций.

Доступные команды:

1 – после введения этой команды пользователь попадает в главное меню игры, где ему доступны различные действия, которые представлены ниже.

0 – данная команда означает начало игры.

9– вывод результатов последних 15 игр на экран

После нажатия клавиши вызова последних 15 игр, вызывается функция fifteen, которая создана для записи игр, путем fopen, fgets и fclose.

5 Руководство пользователя.

Игра “Крестики-нолики”- консольный приложение с сохранением результатов файл. При загрузке программы отобразится заставка, с информацией об авторе, а также организации, которая участвовала в проектировке. После этого пользователь может перейти в главное меню программы

В случае перехода в главное меню:

_____ Меню игры _____
_____ Крестики Нолики _____
_____ ИГРА _____
_____ Крестики-нолики _____
_____ Вы находитесь в главном МЕНЮ игры _____
_____ Крестики-Нолики _____
_____ Ниже описаны комбинации для управления _____
_____ в данной игре _____
_____ для данной игры вам потребуется _____
_____ вводить числа 1-3 на вашей клавиатуре _____
_____ *ввод происходит без клавиши space* _____
_____ после выбора вашей комбинации _____
_____ следует нажать кнопку ENTER _____
_____ 0-начать игру _____
_____ нажмите 9 для вывода 15 последних игр _____
_____ ПГУ 2022 _____

Нажмите 0 если хотите начать:

Если пользователь решит начать игру и подтвердит это нажатием нужной клавиши, появится игровое поле размером 3 на 3 клеток:

	1	2	3
1			
2			
3			

Ваш ход. Введите номер строки и столбца.

Далее нужно указать номер строки и столбца, в которые будет назначен ход игрока, например, 22:

22
1 2 3
1 0
2 X
3

Ваш ход. Введите номер строки и столбца.

После, нужно будет и дальше отображать свой ход, только другому игроку и так далее, до того момента, пока кто-либо не выиграет или случится ничья. Также игроку будет предложено вернуться в главное меню

```

  1 2 3
1|0|0| |
-----
2|x|x|x|
-----
3|0| | |
-----
Победил крестик

Введите цифру 1 для того, чтобы вернуться в главное меню

```

В главном меню пользователь может просмотреть результаты его последних 15 игр, нажав определенную клавишу:

```

_____
_____
_____нажмите 9 для вывода 15 последних игр_____
_____
_____ПГУ 2022_____
Нажмите 0 если хотите начать:

```

```

1Победила дружба
2Победил крестик
3Победил нолик
4Победил нолик
5Победила дружба
6Победил нолик
7Победил крестик
8Победил крестик
9Победил крестик
10Победил крестик
11Победил крестик
12Победил крестик
13Победил крестик
14Победил крестик
15Победил крестик

```

6 Разработка функции на языке Assembler.

В своей работе я использовал ассемблер для проверки игрового поля, чтобы далее можно было определить победителя.

```
inc k
mov eax,k
cmp eax,9
je second
jmp end
second:mov ebx,igra
cmp ebx,0
je done
jmp end
done: mov igra,5
end:
```

Мной были использованы следующие ассемблерные инструкции: инструкция `mov` выполняет копирование содержимого источника в приемник;

инструкция `inc` выполняет сложение содержимого приемника с 1;

инструкция `cmp` выполняет сравнение двух операндов;

`je`— выполняет условный переход, если два операнды равны;

`jmp` – выполняет безусловный переход.

7 Тестирование.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы.

1. Тестирование на корректный запуск программы(откроется титульный лист).

_____ ПГУ _____
_____ КУРСОВАЯ РАБОТА _____
_____ ИГРА _____
_____ Крестики-нолики _____
_____ Выполнил: _____
_____ студент группы 21вв1 _____
_____ Жбанников Д.Н. _____
_____ Принял: _____
_____ Голотенков Н.Г. _____
_____ Пенза 2022 _____
Нажмите 1 для начала игры: ☐

Далее нужно подтвердить выбор о начале игры.

2. Тестирование на корректность запуска меню программы

Меню игры
Крестики Нолики
ИГРА
Крестики-нолики
Вы находитесь в главном МЕНЮ игры
Крестики-Нолики
Ниже описаны комбинации для управления
в данной игре
для данной игры вам потребуется
вводить числа 1-3 на вашей клавиатуре
ввод происходит без клавиши space
после выбора вашей комбинации
следует нажать кнопку ENTER
0-начать игру
нажмите 9 для вывода 15 последних игр
ПГУ 2022
Нажмите 0 если хотите начать: ■

Меню успешно запустилось.

3. Проверка корректности заполнения игрового поля крестиками и ноликами

```

Ваш ход. Введите номер строки и столбца.
11
  1 2 3
1|0| | |
  -----
2| | | |
  -----
3| | | |
  -----
Ваш ход. Введите номер строки и столбца.
21
  1 2 3
1|0| | |
  -----
2|X| | |
  -----
3| | | |
  -----
Ваш ход. Введите номер строки и столбца.
31
  1 2 3
1|0| | |
  -----
2|X| | |
  -----
3|0| | |
  -----
Ваш ход. Введите номер строки и столбца.

```

Как видно, строки и столбцы, которые ввел пользователь, совпадают с игровым полем.

4. Проверка корректности определения победителя

```

  1 2 3
1|0|0|0|
  -----
2|X|X| |
  -----
3| |0|X|
  -----
Победил нолик
Введите цифру 1 для того, чтобы вернуться в главное меню

```

Программа правильно определила победителя, предложив вернуться в меню.

5. Проверка корректности вывода результатов последних игр

1	Победила дружба
2	Победил крестик
3	Победил нолик
4	Победил нолик
5	Победила дружба
6	Победил нолик
7	Победил крестик
8	Победил крестик
9	Победил нолик
10	Победил нолик
11	Победил нолик
12	Победил нолик
13	Победил нолик
14	Победил нолик
15	Победил нолик

В таблице 1 представлена сводная информация по результатам тестирования.

Таблица 1 – Описание поведения программы при тестировании

Описание теста	Ожидаемый результат	Полученный результат
Запуск программы	Открытие титульного листа и меню	Верно
Ввод команд с клавиатуры	Программа корректно принимает команды	Верно
Проверка asm функции сложения строк	Сложение строк производится верно	Верно
Проверка записи в файл	Данные записываются в файл	Верно
Проверка корректности вывода данных из файла	Данные выводятся из файла	Верно
Проверка работоспособности	Сообщения сохраняются в файл	Верно

В результате тестирования программы было выявлено, что программа функционирует успешно.

Заключение.

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Получены знания и навыки по работе с файлами. Получены базовые навыки программирования на языке С и Ассемблера. Изучены способы создания и интеграции с С кодом asm файлов. Получены навыки отладки и тестирования программ.

В рамках данной курсовой работы была написана игра. Данная программа не требует большого количества системных ресурсов и имеет систему команд для удобства и простоты использования.

В дальнейшем планируется усовершенствование программы путем добавления новых структур интерфейса, другое графическое представление, а также будут добавлены новые режимы. Например: игра с полем 5 на 5, игра с компьютером, а также онлайн режим.

Приложение «Листинг программы»

Файл «head.h»

```
#ifndef head_h
#define head_h

extern int a[3][3];

void fifteen();
void nul();
void visual();
void hod(int n);
int proverka();
void nomain();
void menu();
void global();

#endif //Head
```

Файл «noll.cpp»

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"
void nul()
{
    int i,j;

    // пробегаем по всем строкам
    for(i=1; i<=3; i++)
    {
        // пробегаем по всем столбцам
        for (j=1;j<=3;j++)
        {
            a[i][j]=0;
        }
    }
}
```

Файл «visual.cpp»

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"
//int a[3][3];
void visual()
{
```

```

int i,j;
printf ("  1 2 3\n");

// пробегаем по всем строкам
for(i=1; i<=3; i++)
{
    printf ("%d", i);
    // пробегаем по всем столбцам
    for (j=1;j<=3;j++)
    {
        if (a[i][j]==0){printf("| ");};
        if (a[i][j]==1){printf("|0");};
        if (a[i][j]==2){printf("|X");};
    }

    // переходим на следующую строку

    printf ("|\n");
    printf ("_____ \n");
}
}

```

Файл «fifteen.cpp»

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <stdlib.h>
#include "head.h"
void fifteen()
{
    system("cls");
    int k = 0;
    char string[25];
    FILE *stroka = fopen("E:/fifteen.txt", "r");
    do
    {
        k=k+1;
    }
    while(fgets(string,25,stroka)!=NULL);
    fclose(stroka);
    FILE *input = fopen("E:/fifteen.txt", "r");
    for(int i = 1; i < k; i++)
    {
        if((k-i) > 15)
        {
            fgets(string,25,input);
        }
        for(int i = 1;i<16;i++)
        {
            fgets(string,25,input);
            printf("%d",i);
            puts(string);
        }
        fclose(input);
        getch();
    }
}

```

Файл «hod.cpp»

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"

void hod (int n)
{
    int hi; // ход игрока
    int sto; // номер столбца в ходе
    int str; // номер строки в ходе
    int ver=0; // проверка на корректность хода, если нельзя сделать такой ход то 0,
    // если можно то 1
    // запрашиваем ход, пока не будет введен корректный ход
    while (ver==0)
    {
        // ввод хода
        printf ("Ваш ход. Введите номер строки и столбца. \n");
        scanf ("%d", &hi);
        if(hi>3)
        {

        }

        // определяем номер столбца - это последняя цифра в ходе
        sto=hi%10;

        // определяем номер строки хода - это первая цифра
        str=(hi-sto)/10;

        // проверка корректности хода, клетка должна быть пуста и номер должен быть
        // один из номеров таблицы 3 на 3
        if (a[str][sto]==0 && (hi==11 || hi==12 || hi==13 || hi==21 || hi==22 ||
        hi==23 || hi==31 || hi==32 || hi==33 )) (ver=1);
    }

    // если ходил игрок 1 то поле хода 1
    if ( n==1) {a[str][sto]=1;}
    // если ходил игрок 2 то поле хода 2
    if ( n==2) {a[str][sto]=2;}
}

```

Файл «provreka.cpp»

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"
int proverka()
{
    int i;
    int res; // значение функции. Если ни один игрок не выиграл то 0, если выиграл нолик то 1
    // если выиграл крестик то 2
    res=0;
    // проверяем выигрыш первого игрока

```

```

// пробегаем по всем строкам
for (i=1; i<=3; i++)
{
    // проверка все ли в строке нолики
    if (a[i][1]==1 && a[i][2]==1 && a[i][3]==1 ) {res=1;}
}

// пробегаем по всем столбцам
for (i=1; i<=3; i++)
{
    // проверка все ли в столбце нолики
    if (a[1][i]==1 && a[2][i]==1 && a[3][i]==1 ) {res=1;}
}

// проверка диагоналей
if (a[1][1]==1 && a[2][2]==1 && a[3][3]==1 ) {res=1;}
if (a[1][3]==1 && a[2][2]==1 && a[3][1]==1 ) {res=1;}

// проверяем выигрыш второго игрока
// пробегаем по всем строкам
for (i=1; i<=3; i++)
{
    // проверка все ли в строке крестики
    if (a[i][1]==2 && a[i][2]==2 && a[i][3]==2 ) {res=2;}
}

// пробегаем по всем столбцам
for (i=1; i<=3; i++)
{
    // проверка все ли в столбце крестики
    if (a[1][i]==2 && a[2][i]==2 && a[3][i]==2 ) {res=2;}
}

// проверка диагоналей
if (a[1][1]==2 && a[2][2]==2 && a[3][3]==2 ) {res=2;}
if (a[1][3]==2 && a[2][2]==2 && a[3][1]==2 ) {res=2;}
return res;
}

```

Файл «nomain.h»

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"
void nomain()
{
    int start;
    int k=0;
    system("cls");
    int igra;

    nul();
    // отображаем игровое поле
    visual();
    igra=0;

    // пока никто не победил продолжается игра

    while (igra==0)
    {
        // ход 1 го игрока нолика

```

```

        hod(1);
        // отображаем игровое поле
        visual();
        // проверка на победу первого игрока
        igra=proverka();
        // если первый игрок не победил , то ходит второй игрок

        _asm{
            inc k
            mov eax,k
            cmp eax,9
            je second
            jmp end
second:    mov ebx,igra
            cmp ebx,0
            je done
            jmp end
done:     mov igra,5
end:
        }

        if(igra==0)
        {
            // ход второго игрока
            hod(2);
            // отображаем игровое поле
            visual();
            // проверка на победу второго игрока
            igra=proverka();
            k+=1;
        }
    }

    // отображение результатов игры
    FILE *output = fopen("E:/fifteen.txt","a");
    if (igra==1)
    {
        printf ("Победил нолик\n");
        fprintf(output,"Победил нолик\n");
    }
    if (igra==2)
    {
        printf ("Победил крестик\n");
        fprintf(output,"Победил крестик\n");
    }
    if (igra==5)
    {
        printf ("Победила дружба\n");
        fprintf(output,"Победила дружба\n");
    }
    fclose(output);

    printf("\n
    в главное меню\n
    scanf("%d", &start);
    if (start == 1)
    {
        menu();
    }
    getch();
}
Введите цифру 1 для того, чтобы вернуться
");

```

Файл «menu.cpp»

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <stdlib.h>
#include "head.h"

void menu()
{
    int startik;
    setlocale(LC_ALL, "Rus");
    printf (
        _____ \n");
    printf (
        "| _____ Меню игры _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ Крестики Нолики _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ ИГРА _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ Крестики-нолики _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ Вы
находитесь в главном МЕНЮ игры _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ Крестики-Нолики _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ Ниже
описаны комбинации для управления _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ в данной игре _____ | \n");
    printf (
        "| _____ для
данной игры вам потребуется _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ вводить числа 1-3 на вашей клавиатуре _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ *ввод
происходит без клавиши space* _____ | \n");
    printf (
        "| _____ | \n");
    printf (
        "| _____ после
выбора вашей комбинации _____ | \n");
    printf (
        "| _____ | \n");

```

```

printf (
"|_____следует нажать кнопку ENTER_____|\n");
printf (
"|_____|\n");
printf (
"|_____0-начать игру_____|\n");
printf (
"|_____|\n");
printf (
"|_____|\n");
printf (
"|_____нажмите
9 для вывода 15 последних игр_____|\n");
printf (
"|_____|\n");
printf (
"_____ПГУ 2022_____ \n");

printf ("Нажмите 0 если хотите начать: ");
scanf("%d",&startik);
if(startik==0)
{
nomain();
}
if(startik==9)
{
fifteen();
}
getch();
}

```

Файл «global.cpp»

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"

void global()
{
    int startik;
    setlocale(LC_ALL, "Rus");
    printf (
"_____ \n");
    printf (
"|_____ПГУ_____|\n");
    printf (
"|_____|\n");
    printf (
"|_____КУРСОВАЯ РАБОТА_____|\n");
    printf (
"|_____|\n");
    printf (
"|_____ИГРА_____|\n");
    printf (
"|_____|\n");
    printf (
"|_____Крестики-нолики_____|\n");
    printf (
"|_____|\n");
    printf (
"|_____Выполнил:|\n");
    printf (
"|_____|\n");
}

```

```

printf (
"| _____студент группы 21вв1|\n");
printf (
"| _____|\n");
printf (
"| _____Жбанников Д.Н.|\n");
printf (
"| _____|\n");
printf (
"| _____Принял:|\n");
printf (
"| _____|\n");
printf (
"| _____Голотенков Н.Г.|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____|\n");
printf (
"| _____Пенза 2022|\n");
printf (
"| _____|\n");
printf (
"_____|\n");

printf ("Нажмите 1 для начала игры: ");
scanf("%d",&startik);
if(startik==1)
{
menu();
}

getch();
}

}

```

Файл «baza.cpp»

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "head.h"

int a[3][3] = {(0,0,0),(0,0,0),(0,0,0)};
void main()
{
global();
}

```

