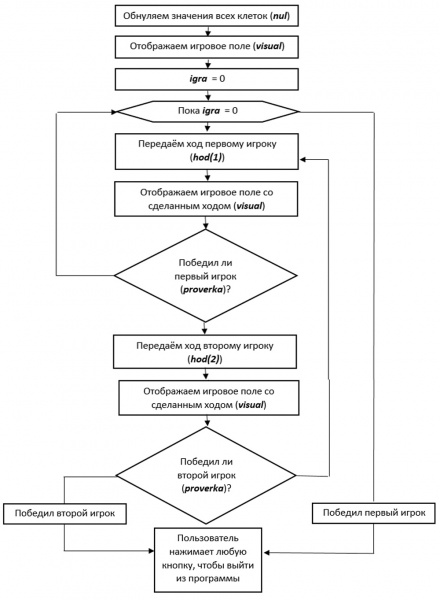
**Игра крестики нолики на Си**

Задача заключается в реализации игры крестики-нолики на языке программирования Си++.  Программа должна создавать поле для игры в крестики-нолики. Игроки поочерёдно вводят номер строки и столбца, в котором они хотят сделать ход. Игра проходит таким образом до тех пор, пока кто-либо из игроков не победит или не произойдёт ничья. Если клетка уже занята, то программа не должна дать сделать некорректный ход. Если кто-либо победит, программа должна уведомить об этом, после чего пользователь должен нажать кнопку, чтобы выйти из программы.

Краткое описание алгоритма программы **крестики нолики на си.**При запуске программы показывается игровое поле, после чего первый игрок должен ввести номер строки и столбца клетки, на которой он хочет сделать ход. Затем такие же действия должен совершить второй игрок. Программа считывает номер строки и столбца, введённые пользователем, проверяет, занято ли выбранное место, если занято, то предлагает сделать ход заново, если нет, то ход будет сделан. Если кто-либо из игроков победил, программа выводит сообщение о том, кто из игроков победил, после чего пользователь нажимают любую кнопку, чтобы выйти из программы.

[](http://itrobo.ru/upload/wysiwyg/2b5d739649f3a2298e90af6539333f6c.jpg)

Переменные и массивы. Массив a[i][j] отвечает за номер строки и столбца каждой клетки. i – номер строки клетки, j – номер столбца клетки. Подробно о работе с [массивами в Си](http://itrobo.ru/programmirovanie/osnovy-si/matricy-v-si.html)

Если a[i][j] =0 то данная клетка свободна

Если a[i][j] =1 то данная клетка занята ноликом

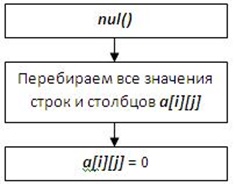
Если a[i][j] =2 то данная клетка занята крестиком

Переменные igra и res отвечают за состояние игры, если эти переменные равны нулю, то пока никто не победил, если 1 – победил нолик, если 2 – победил крестик. В переменную hi записывается ход игрока, например, «12» - первая строка, второй столбец. str и sto отвечают за номер выбранных строки и столбца. ver – проверяет возможность хода, если ход невозможен – 0, если возможен – 1.

В программе присутствует четыре процедуры. Подробно  о [процедурах в Си.](http://itrobo.ru/programmirovanie/osnovy-si/procedury-i-funkcii-v-si.html)

nul() обнуляет находящиеся элементы каждой клетки на поле, это нужно для начала игры. Если клетка была пустой, то процедура оставит её пустой, если клетка была занята крестиком или ноликом, процедура уберёт их оттуда.

Алгоритм процедуры

[](http://itrobo.ru/upload/wysiwyg/b4cb7c115a8cb0ddf8f88cda34c3d2fa.jpg)

visual() предназначена для визуализации поля при каждом ходе. Если клетка пуста, то ничего в ней не нарисуется, если на клетке сделали ход, то процедура нарисует этот ход крестик или нолик.

Алгоритм процедуры визаулизации поля

[](http://itrobo.ru/upload/wysiwyg/e61057a929e3d4466e1fdbd17a7b8671.jpg)

Процедура hod(n). Процедура запрашивает номер строки и столбца в клетки, в которую ходит игрок и назначает на выбранную клетку нолик или крестик.  n – номер игрока, который должен ходить в данный момент. Если n = 1 то ходит нолик, если n = 2, то ходит крестик.

Для  проверки 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);

Алгоритм процедуры ход.

[](http://itrobo.ru/upload/wysiwyg/74ba84ccf30501be3a8237c1dfdbbcdd.jpg)

proverka() при каждом ходе проверяет, не победил ли какой-либо игрок, и возвращает одно из возможных значений. Если proverka() = 0, то никто пока не победил или случилась ничья, если proverka() = 1, то победил нолик, если 2 – крестик.

При проверке мы проверяем в цикле for  все строки и ищем есть в какой либо строке  все нолики

// 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;}

 }

Подробно о [цикле for в Си](http://itrobo.ru/programmirovanie/osnovy-si/cikly-v-si.html)

Аналогично идет проверрка по всем столбцам и проверка диагоналей

 // 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;}

Алгоритм проверки

[](http://itrobo.ru/upload/wysiwyg/8a1893f46195d451457d29985b0a8fb9.jpg)

Полный код программы.

#include <stdio.h>  
#include <conio.h>

int a[3][3];// объявляем  текущее поле  3 строки и 3 столбца. если  нет ничего то поле 0, если нолик,  то  1, если крестик то 2

// обнуление поля

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;  
  }    
 }  
}

// вывод  поля на экран

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("|O");};  
      if (a[i][j]==2){printf("|X");};  
  }

// переходим на следующую строку

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

// Ход игрока n - номер игрока На выходе: 0 -

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);

// 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;}   
}

// Проверка  Если ни один игрок не выиграл то 0, если выиграл нолик то 1 если выиграл крестик то 2  
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;  
}

main()  
{  
int igra;// состояние игры если победил нолик то 1 , если победил крестик то 2 если никто пока не победил то 0  
// обнуляем очищаем поле  
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

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

}

// отображение результатов игры  
if (igra==1){printf ("Победил нолик\n");};  
if (igra==2){printf ("Победил крестик\n");};  
getch();  
}

Задание для самостоятельной работы.

Расширьте игровое поле до размеров 5×5 клеток.