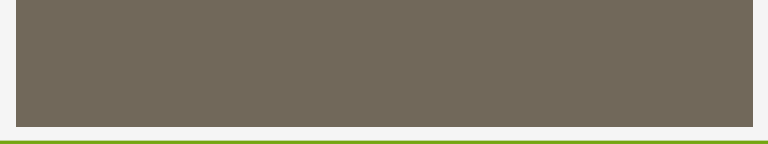




Задачи

Изготвил:
гл.ас. д-р Нора Ангелова



Да се напише програма, която въвежда координатите на две полета от шахматна дъска и проверява може ли с един ход да се стигне от първото поле до второ с:

- пешка
- топ
- офицер

a)

```
#include <iostream.h>
int main()
{ // координатите на първата точка
  int x1, y1;
  do
  { cout << "x1= ";
    cin >> x1;
    cout << "y1= ";
    cin >> y1;
  } while(x1 < 1 || x1 > 8 || y1 < 1 || y1 > 8);
  // координатите на втората точка
  int x2, y2;
  do
  { cout << "x2= ";
    cin >> x2;
    cout << "y2= ";
    cin >> y2;
  } while(x2 < 1 || x2 > 8 || y2 < 1 || y2 > 8);
  // фигурата е пешка
  // клетките достъпни от първата точка са:
  // (x1 - 1, y1 + 1), (x1, y1 + 1), (x1 + 1, y1 + 1)
  // (x1 - 1, y1 - 1), (x1, y1 - 1), (x1 + 1, y1 - 1)
  if((x1 - 1 == x2 || x1 == x2 || x1 + 1 == x2) &&
     (y1 - 1 == y2 || y1 + 1 == y2))
    cout << "Втората точка е достижима с пешка.\n";
  else
    cout << "Втората точка не е достижима с пешка.\n";
  return 0;
}
```

б) Решението е аналогично на това от а), но последният условен оператор

$x1 == x2 \ || \ y1 == y2.$


```

int main()
{ // координатите на първата точка
  int x1, y1;
  do
  { cout << "x1= ";
    cin >> x1;
    cout << "y1= ";
    cin >> y1;
  } while(x1 < 1 || x1 > 8 || y1 < 1 || y1 > 8);
  // координатите на втората точка
  int x2, y2;
  do
  { cout << "x2= ";
    cin >> x2;
    cout << "y2= ";
    cin >> y2;
  } while(x2 < 1 || x2 > 8 || y2 < 1 || y2 > 8);
  // фигурата е пешка
  // клетките достъпни от първата точка са:
  // (x1 - 1, y1 + 1), (x1, y1 + 1), (x1 + 1, y1 + 1)
  // (x1 - 1, y1 - 1), (x1, y1 - 1), (x1 + 1, y1 - 1)
  if((x1 - 1 == x2 || x1 == x2 || x1 + 1 == x2) &&
      (y1 - 1 == y2 || y1 + 1 == y2))
    cout << "Втората точка е достижима с пешка.\n";
  else
    cout << "Втората точка не е достижима с пешка.\n";
  return 0;
}

```

б) Решението е аналогично на това от а), но последният условен оператор
 $x1 == x2 \text{ || } y1 == y2$.


в) Решението е аналогично на това от а), но последният условен оператор
 $x1 - y1 == x2 - y2 \text{ || } x1 + y1 == x2 + y2$.



Да се напише програма, която проверява дали сумата от цифрите на положително число е кратна на 3.

Решение:

```
#include <iostream.h>
int main()
{ int n;
  do
  { cout << "Въведете положително цяло число ";
    cin >> n;
  } while(n <= 0);
  // намиране на сумата от цифрите на n
  int s = 0;
  int k = n;
  while(k > 0)
  { s = s + k % 10;
    k = k / 10;
  }
  // проверка дали сумата от цифрите се дели на 3
  if(s % 3 == 0) cout << "Сумата от цифрите на " << n
    << " се дели на 3. \n";
  else cout << "Сумата от цифрите на " << n
    << " не се дели на 3. \n";
  return 0;
}
```



Нека x е число от редицата на Фибоначи. Да се напише програма, която намира поредния номер на числото в редицата от числа на Фибоначи.


```
#include <iostream.h>
int main()
{ int a = 0, // 0-вото число
  b = 1, // 1-вото число
  x = 0, // дадено число
  n = 0, // пореден номер на даденото число
  tmp = 0;
  cout << "Въведете число на Фибоначи: ";
  cin >> x;
  // намиране на мястото на числото в редицата
  while(a < x)
  { tmp = b;
    b = a + b;
    a = tmp;
    n++;
  }
  if(a != x) cout << x << " не е число на Фибоначи\
else cout << "Поредният номер на " << x
        << " е: " << n << "\n";
  return 0;
}
```


1 2 3 4 4 3 2 1

2 3 4 4 3 2

3 4 4 3

4 4

3 4 4 3

2 3 4 4 3 2

1 2 3 4 4 3 2 1

```
#include <iostream.h>
#include <iomanip.h>
int main()
{ int i, j;
  // Извеждане на първите 4 реда
  for(i = 1; i <= 4; i++)
  { cout << setw(2*(i-1)+1) << ' ';
    for(j = i; j <= 4; j++)
      cout << j << " ";
    for(j = 4; j >= i; j--)
      cout << j << " ";
    cout << endl;
  }
  // Извеждане на последните 3 реда
  for(i=3; i >= 1; i--)
  { cout << setw(2*(i-1)+1) << ' ';
    for(int j = i; j <= 4; j++)
      cout << j << " ";
    for(j = 4; j >= i; j--)
      cout << j << " ";
    cout << endl;
  }
  return 0;
}
```