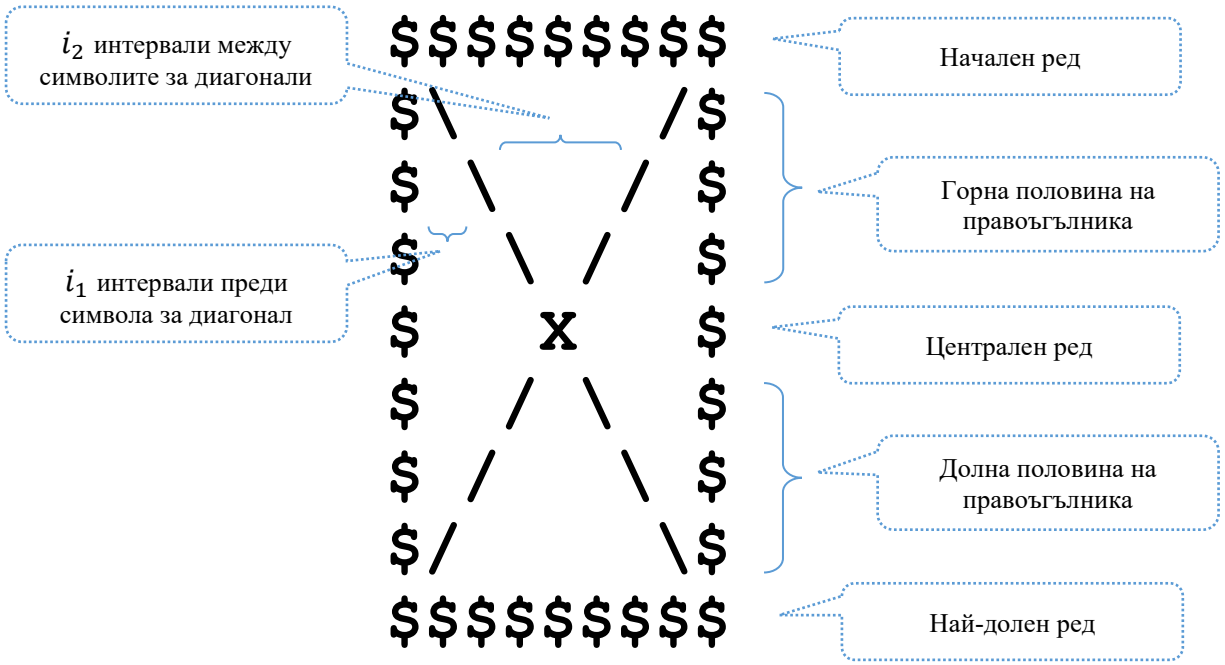
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА НАРИСУВАЙ МИ ПРАВОЪГЪЛНИК!

Да разгледаме един правоъгълник, нарисуван със знака **$**. Картинката на правоъгълника се състои от 5 части: горен ред, редове с горната половина на правоъгълника (не включват най-горния ред и централния ред, ако го има), централен ред (има го, ако N е нечетно), редове с долната половина на правоъгълника (не включват централния ред, ако го има, и най-долния ред) и най-долния ред на правоъгълника. Да разгледаме как се програмира изписването на всяка част.



Най-горен ред

В тяло на цикъл отпечатваме еднократно символа, с който рисуваме рамката на правоъгълника. Изпълняваме тялото на цикъла N пъти. След това минаваме на нов ред.

Редове с горната половина на правоъгълника

Броят на тези редове е $N/2-1$. Организираме цикъл, в тялото на който изписваме поредния от редовете. Всеки ред съдържа:

- Символа, с който се изписва рамката. Изписваме го.
- i_1 интервали до символа, за изписване на лявонаклонения диагонал. Изписваме ги с помощта на цикъл, в чието тяло изписваме по един интервал, а тялото се повтаря i_1 пъти.
- Символа, с който се изписва лявонаклонения диагонал. Изписваме го.
- i_2 интервали между символите за изписване на диагоналите. И тях изписваме с помощта на цикъл.
- Символа, с който се изписва дяснонаклонения диагонал. Изписваме го.
- Нови i_1 интервали до десния край на рамката. Изписваме ги с цикъл, както бе обяснено по-горе.
- Символа, с който се изписва рамката. Изписваме го и минаваме на нов ред.

Когато организираме цикъла за изписване на редовете от горната половина на правоъгълника съобразяваме, че:

- i_1 за всеки следващ ред увеличава стойността си с 1, като започва от 0 (за най-горния ред от групата редове)
- i_2 за всеки следващ ред намалява стойността си с 2, като започва от N – 4 (за най-горния ред от групата редове)

Централен ред

Изписваме го, само ако N е нечетно! Редът се състои от символите за изписване на рамката на правоъгалника (по един в двата края на реда), по (N-3)/2 интервали от двете страни на централния X в реда и самия централен символ X. Интервалите преди и след централния символ изписваме отново с цикли.

Редове с долната половина на правоъгълника

И техният брой е $N/2-1$. Постъпваме както бе описано при редовете от горната половина на правоъгълника като съобразяваме няколко важни разлики:

- В реда символът, с който се изписва дяснонаклонения диагонал, е преди символа за лявонаклонения диагонал.
- i_1 за всеки следващ ред намалява стойността си с 1, като започва от $N/2-2$ (за най-горния ред от групата редове)
- i_2 за всеки следващ ред увеличава стойността си с 2, като започва от 0 при четно N или от 1 при нечетно N (за най-горния ред от групата редове)

Най-долен ред

Еднакъв е с най-горния. Изписваме го по същия начин.

Някои тънкости при реализацията

При изчисляване на броя редове за горната и долната половина на правоъгълника ($N/2-1$) трябва да приложим целочислено делене на 2. Така формулата смята правилно броя редове и при четно, и при нечетно N.

Ако правилно организираме цикъла за изписване на горната половина редове на правоъгълника, стойността на i_1 ще бъде актуализирана при последното изпълнение на тялото на цикъла до правилната стойност за брой интервали преди централния X в централния ред – можем да я наследим директно в частта за изписване на централния ред.

Ако N=1, правоъгълникът се състои само от горния ред, в който има само един символ. В този случай след изписване на началния ред прекратяваме програмата предсрочно.

Сорс-кодът на програмата

```
#include <iostream>
using namespace std;

int main() {
  int n,i,j,i1,i2;
  char c;
  cin >> n >> c;

  /// Рисува горната страна
  for (i=0; i<n ; i++) cout << c; cout << endl;
  if (n==1) return 0;
```

```
  /// Рисува горната половина от правоъгълника
  for (j=0, i1=0, i2=n-4; j<n/2-1; j++, i1++, i2-=2){
    cout << c;
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << '\\';
    for (i=0; i<i2 ; i++) cout << ' ';
    cout << '/';
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << c << endl;
  }
  /// Рисува "централния" ред, ако има такъв
  if (n%2) {
    cout << c;
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << 'X';
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << c << endl;
  }
  /// Рисува долната половина от правоъгълника
  for (j=0, --i1, i2=n%2; j<n/2-1; j++, i1--, i2+=2){
    cout << c;
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << '/';
    for (i=0; i<i2 ; i++) cout << ' ';
    cout << '\\';
    for (i=0; i<i1 ; i++) cout << ' ';
    cout << c << endl;
  }
  /// Рисува долната страна
  for (i=0; i<n ; i++) cout << c; cout << endl;

  return 0;
}
```

Допълнителни улеснения

Прави впечатление, че на много места се налага да програмираме отпечатване на няколко поредни интервала. Правим го с цикъл. Копираме първото срещане на този цикъл на останалите места, където се изписват интервали, като след копирането коригираме броя изпълнения на тялото (обикновено чрез редактиране на управляващото условие на цикъла).

Може да постигнем желания ефект, ако се възползваме от възможностите за форматиран изход на извежданите данни. Може да се укаже при отпечатването всяка данна (в нашия случай символ) да се разположи в място (поле) с посочена широчина (в брой символи). Ако широчината на данната е по-малка от заделеното поле за отпечатването ѝ, тя се разполага в дясната част на полето, а свободната част от полето вляво от данната се запълва автоматично с интервали. Задаването на поле за отпечатване на данна се извършва като се накара `cout` да отпечата псевдоданната `setw(`*`число`*`)` преди данната за отпечатване. Псевдоданната `setw(`*`число`*`)` се нарича манипулатор и, за да я ползвате, в началото на програмата трябва да добавите ред за включване на библиотеката с манипулаторите:

```
#include <iomanip>
```

При използването на `setw(`*`число`*`)` трябва да подмените *`число`* с числова константа или израз, който показва колко символа да е широко задаваното поле за отпечатване на следващата истинска данна за отпечатване – нещо такова:

```
cout << "Imam" << setw(5) <<  25 << setw(15) << "uchenici" << endl;
```

ще отпечата:

```
Imam    25        uchenici
```

В библиотеката iomanip има още много средства за оформяне (форматиране) на извежданите данни – бързо накарайте ръководителите Ви да Ви ги разкажат! А ето и вариант на програмата ни, в който са използвани някои от възможностите за форматиране на изходните данни:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
  int n,i,j,i1,i2;
  char c;
  cin >> n >> c;

  /// Рисува горната страна
  for (i=0; i<n ; i++) cout << c; cout << endl;
  if (n==1) return 0;

  /// Рисува горната половина от правоъгълника
  for (j=0, i1=0, i2=n-4; j<n/2-1; j++, i1++, i2-=2)
    cout << c << setw(i1+1) << '\\' << setw(i2+1) << '/' << setw(i1+1) << c << endl;

  /// Рисува "централния" ред, ако има такъв
  if (n%2) cout << c << setw(i1+1) << 'X' << setw(i1+1) << c << endl;

  /// Рисува долната половина от правоъгълника
  for (j=0, --i1, i2=n%2; j<n/2-1; j++, i1--, i2+=2)
    cout << c << setw(i1+1) << '/' << setw(i2+1) << '\\' << setw(i1+1) << c << endl;

  /// Рисува долната страна
  for (i=0; i<n ; i++) cout << c; cout << endl;

  return 0;
}
```

Още един подход за решаване на задачата

Да погледнем към решавания проблем от друга гледна точка.

Картинката на правоъгълника се състои от *N* реда с по *N* символи. С помощта на два вложени цъкъла изписваме всеки от символите (с външния цикъл се изписва ред от картинката, а с вътрешния цикъл – пореден символ в реда). Преди да изпишем поредния символ преценяваме, според мястото където се намира, какъв да е символът. Ако се намира по границите на картинката – изписваме знака *patern*, ако е по диагоналите – изписваме съответния знак за диагонал. На всички останали места изписваме интервал. Критерий къде се намира изписвания символ са номерът на реда и позицията в този ред в картинката на правоъгълника:

- За символи по рамката на правоъгълника номерата на ред, или пък позициите в реда, са или 0, или *N*-1 (броенето на редовете и позициите в тях е от 0);
- За символи по лявонаклонения диагонал номерата на ред и позиция в реда съвпадат. Не трябва да забравяме, че при нечетно *N* и при централен ред от картинката трябва да изобразим **X** вместо \ ;

– Символ за дяснонаклонения диагонал изписваме, ако сумата от ред и позиция в този ред е *N*-1 (броенето на редовете и позициите в тях е от 0).

Сорс-кодът на алтернативния вариант

```
#include <iostream>
using namespace std;

int main() {
  int i,j,n,m;
  char c;
  cin >> n >> c;
  m = n-1;
  for (i=0; i<n ; i++) {
    for (j=0; j<n; j++){
      if (i==0 or j==0 or i==m or j==m) cout << c;/// По рамката
      else if (i==j) {                           /// По лявонаклонения диагонал
        if (i==n/2 and n%2==1) cout << 'X';      ///   - в центъра
        else cout << '\\';                       ///   - на друго място
      }
      else if (i+j == m) cout << '/';            /// По дяснонаклонения диагонал
      else cout << ' ';                          /// Празно място в картинката
    }
    cout << endl;
  }
  return 0;
}
```

Автор: Евгений Василев