АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
КАМЪНИ

Задачата е традиционна игра. Нека с номер *k* означим позицията от играта, при която на масата са останали *k* камъка. Позицията *k* ще наричаме **печеливша**, ако играчът, който е на ход, при правилна стратегия, може да спечели играта. Позицията ще наричаме **губеща**, ако играчът, който е на ход, задължително ще загуби играта при правилна стратегия на противника. Очевидно позиция 0 (нула камъка, останали на масата) е губеща за играча, който е на ход. Също е очевидно, че позиция 1 (1 камък, останал на масата) е печеливша за играча, който е на ход, тъй като той ще вземе този камък.
По-нататък това дали позиция *k* е печеливша или губеща се определя по правилото:

- позиция *k* е губеща, ако всеки възможен ход от нея води в печеливша позиция;
- позиция *k* е печеливша, ако има ход, който води в губеща позиция.

Това правило позволява да се определи последователно свойството „печеливша" или „губеща" за всяка позиция от 1 до 1000000 (напомняме, че на масата има най-много 1000000 камъка) и тази информация да се запише в масив. При въвеждане на данните за серия от игри, програмата само трябва да проверява каква е началната позиция за всяка игра според началния брой на камъните в нея: ако началната позиция е „печеливша", то за тази игра се извежда 1, а ако е „губеща" - 2.

Реализация

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

bool positions[1000001];
char ans[101];

void fillpositions()
{
 int i,k;
 positions[0] = false;
 positions[1] = true;
 positions[2] = true;
 for (i=3;i<=10000;i++)
 {
  k = i%3;
  switch (k)
  {
   case 0:
    if (positions[i-1] && positions[i-2])
     positions[i] = false;
    else
     positions[i] = true;
    break;
   case 1:
    if (positions[i-1] && positions[i-3])
     positions[i] = false;
    else
     positions[i] = true;
```

```
      break;
    case 2:
      if (positions[i-1] && positions[i-2] && positions[i-3])
        positions[i] = false;
      else
        positions[i] = true;
      break;

   }
  }
}



int main()
{
 int n,m,k;
 fillpositions();
 cin >> n ;
 for (k=1;k<=n;k++)
 {
   cin >> m;
   if  (positions[m])
     ans[k] = '1';
   else
     ans[k] = '2';
 }
 for (k=1;k<=n;k++)
   cout << ans[k];
 cout << endl;

 return 0;
}
```

Автор: Руско Шиков