**АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА**
**САМОЛЕТИ**

Тази задача се решава като се симулират полетите на двата самолета и по време на всеки полет се определя разстоянието, което се изминава, като и наличието на разминавания. Когато поредният самолет кацне, трябва да се проверява дали на същото летище в момента не се намира другият самолет и, ако е там, се увеличава броят на засичанията. При симулацията могат да се използват формули и по-сложни условия за по-бързо определяне на летището, на което трябва да кацне самолетът и за това, дали, при съответния полет, той ще прелети над летище, на което е кацнал другият самолет, но подобен подход при малките крие рискове от грешки. Поради това, в реализацията по-долу, симулацията е направена най-елементарно – симулират се преходите от летище към летище. При ограниченията, които са дадени, няма нужда от хитрини за бързодействие – важно е да се работи вярно.

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

const int nmax=1000;

int n,ap1,ap2,d;

bool p1[nmax+1];          //p1[i]=true, ако самолет 1 вече е кацал на летище с номер i;
bool p2[nmax+1];          //p2[i]=true, ако самолет 2 вече е кацал на летище с номер i;

int cap1=1;    // летище, на което е кацнал в момента самолет 1; първоначално на летище 1
int cap2=1;    // летище, на което е кацнал в момента самолет 2; първоначално на летище 1
int countap=1;            // брой на засичанията; първоначално 1 - на летище 1
int countpass=0;          // брой на разминаванията; първоначално 0
long long td1=0;          // разстояние, изминато от самолет 1; първоначално 0
long long td2=0;          // разстояние, изминато от самолет 2; първоначално 0

int main()
{
 int i;
 int tap;       // номер на текущо летище, над което, по време на полет, се намира самолетът
 int nap;     // над колко летища е прелетял, без да кацне, самолетът по време на тек. полет
 bool fly;   // индикация, че самолетът трябва да лети

 cin >> n >> d >> ap1 >> ap2;

// Първоначално и двата самолета са посещавали само летище 1
 for (i=1;i<=n;i++)
 {
  p1[i]=false;
  p2[i]=false;
 }

 p1[1]=true;
 p2[1]=true;
```

```
// Симулиране на (n-1) полета за всеки от двата самолета
  for (i=1;i<n;i++)
  {
    // за първи самолет

    nap=0;                  // В началото на полета не е прелетял без да кацне над нито едно
летище
    tap=cap1;          // При излитането се намира над летището, от което е излетял
    fly=true;          // Самолет 1 излита :-)

    // След излитането започва следният цикъл:
    do
    {
      tap++;                        // Самолет 1 долита до следващото летище.
      if (tap>n) tap=tap-n;
      td1=td1+d;                // Увеличава се разстоянието, което е изминал самолет 1 с d.
      if ((nap>=ap1) && (!p1[tap]))     // Проверява се трябва ли самолет 1 да кацне на
                                        // летището(т.е. дали е прелетял поне над ap1 летища
                                        // и все още не е кацал на това летище)
        fly=false;                  // Ако трябва да каца, слага индикация за край на полета
      else                          // Полетът ще продължи
      {
        nap++;                  // Увеличава се броят на летищата, над които е прелетял самолет
                                   1
                                // по време на текущия полет
        if (tap == cap2)    // Проверява се дали самолет 2 в момента е кацнал на това летище
          countpass++;      // ако е, се увеличава броят на разминаванията с 1.
      }
    }
    while (fly);

    // След като самолет 1 кацне, се прави следното

    cap1=tap;                   // Сменя се летището, на което в момента е кацнал самолет 1
    p1[cap1]=true;              // Отбелязва се, че самолет 1 вече е посетил летище cap1
    if (cap1 == cap2)           // Ако на същото летище в момента е кацнал и самолет 2,
      countap++;                // то се увеличава броят на засичанията.

    // за втори самолет

    nap=0;                  // В началото на полета не е прелетял без да кацне над нито едно
летище
    tap=cap2;          // При излитането се намира над летището, от което е излетял
    fly=true;          // Самолет 2 излита :-)

    // След излитането започва следният цикъл:
    do
    {
      tap--;                        // Самолет 2 долита до следващото летище.
      if (tap<1) tap=tap+n;
      td2=td2+d;                // Увеличава се разстоянието, което е изминал самолет 2 с d.
      if ((nap>=ap2) && (!p2[tap]))     // Проверява се трябва ли самолет 2 да кацне на
```

```
                                        // летището (т.е. дали е прелетял поне над ap2
                                        летища // и все още не е кацал на това летище)
      fly=false;              // Ако трябва да каца, слага индикация за край на полета
     else                     // Полетът ще продължи
     {
      nap++;                 // Увеличава се броят на летищата, над които е прелетял самолет
                             2 // по време на текущия полет
      if (tap == cap1)         // Проверява се дали самолет 1 в момента е кацнал на това
летище
       countpass++;          // ако е, се увеличава броят на разминаванията с 1.
     }
   }
   while (fly);

   // След като самолет 2 кацне, се прави следното

   cap2=tap;                 // Сменя се летището, на което в момента е кацнал самолет 2
   p2[cap2]=true;            // Отбелязва се, че самолет 2 вече е посетил летище cap2
   if (cap2 == cap1)         // Ако на същото летище в момента е кацнал и самолет 1,
    countap++;               // то се увеличава броят на засичанията.
  }
 cout << td1 << " " << td2 << " " << countap << " " << countpass << endl;
 return 0;
}
```

***Автор:*** *Руско Шиков*