

Задача С3. Числа

Дадени са 3 двуцифрени числа d_1, d_2, d_3 .

Напишете програма **NUMBER**, която намира броя M на n -цифрените числа $a_1 a_2 \dots a_n$, за които всеки две съседни цифри $a_i a_{i+1}$ образуват двуцифрено число (без водеща нула), което се дели на някое от числата d_1, d_2, d_3 .

Данните се въвеждат от стандартния вход, където на един ред са записани последователно числата n, d_1, d_2, d_3 , разделени с по един интервал.

Търсеният брой M да се изведе на стандартния изход.

Ограничения:

а) $1 < n < 20$ (за 50% от тестовите примери: $1 < n < 10$)

б) за всички тестови примери $M < 10000$.

Пример

Вход

3 13 28 34

Изход

15

Обяснение на примера:

134, 139, 265, 268, 284, 391, 526, 528, 565, 568, 652, 656, 684, 784, 913

Решение:

Следва пълният текст на програма, която решава задачата:

```
// Идея: Рекурсивно генериране на всички числа
#include <iostream>
using namespace std;
int n, d1, d2, d3;
int a[22];
int M=0;
bool OK(int a, int b)
// проверява дали две цифри a и b могат
// да бъдат една след друга в числото
// без водеща нула
{ if(a==0) return false;
  int x = 10*a + b;
  if(x%d1==0) return true;
  if(x%d2==0) return true;
  if(x%d3==0) return true;
  return false;
}
void Gen(int k) // избира k-тата цифра
{ for(int i=0; i<10; i++)
  { if(OK(a[k-1],i))
    { a[k]=i;
      if(k<n) Gen(k+1); // рекурсивно търси (k+1)-вата цифра
      else M++;        // намерено е още едно от търсените числа
    }
  }
}
int main()
{ cin >> n >> d1 >> d2 >> d3;
  for(int i=1; i<10; i++)
  { a[1]=i; // първа цифра: 1,2,...,9
    Gen(2); // генерира останалите цифри
  }
  cout << M << endl;
  return 0;
}
```