Mars

Məlumdur ki, kosmosa ilk çatan Fironlar olub. Onlar I Tutmos planetinə (hal-hazırda Mars kimi tanınır) ayaq basan ilk kosmik gəmini buraxdılar. Planetin səthi hər bir xanası quru və ya su olan kvadrat xanalardan ibarət $(2n+1) \times (2n+1)$ ölçülü düzbucaqlı kimi modelləşdirilə bilər. i-ci sətir, j-ci sütunda $(0 \le i, j \le 2 \cdot n)$ yerləşən xana quru olarsa, bu $s[i][j] =$`'1'`, su olarsa $s[i][j] =$`'0'` ilə işarələnir.

İki quru xana o zaman əlaqəli sayılır ki, onlar arasında qonşu quru xanalardan keçməklə getmək olsun. Bir kənarı ortaq olan iki xana qonşu xanalar sayılır. Planetdə ada elə maksimal quru xanalar çoxluğuna deyilir ki, bu çoxluqdakı istənilən iki xana əlaqəli olsun.

Kosmik gəminin vəzifəsi planetdəki adaların sayını hesablamaq idi. Lakin kosmik gəminin qədim kompüterinə görə bu tapşırıq asan deyildi. Kompüterdə məlumatı $(2n+1) \times (2n+1)$ ölçülü massiv şəklində saxlayan h yaddaşı var idi. Burada massivin hər bir elementi 100 uzunluqlu, hər bir simvolu ya `'0'` (ASCII 48) ya da `'1'` (ASCII 49) olan ikili sətir saxlaya bilər. Başlanğıcda, hər bir yaddaş xanasının ilk biti düzbucaqlının uyğun xanasının vəziyyətini saxlayır, yəni ki, $h[i][j][0] = s[i][j]$ $(0 \le i, j \le 2 \cdot n)$. h-ın digər bütün bitləri başlanğıcda `'0'`-dır (ASCII 48).

Yaddaşda saxlanan məlumatları emal etmək üçün, kompüter yaddaşın yalnız 3×3 ölçülü hissəsinə baxa və həmin hissənin yuxarı sol xanasındakı dəyərin üzərinə yaza bilər. Daha dəqiq desək, kompüter $h[i..i+2][j..j+2]$ $(0 \le i, j \le 2 \cdot (n-1))$ xanalarında saxlanan dəyərlərə baxa və $h[i][j]$ xanasının üzərinə yeni dəyər yaza bilər. Bu prosesi sonradan (i, j) **xanasını emal** adlandıracağıq.

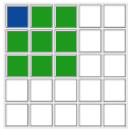
Kompüterin məhdudiyyətlərinin öhdəsindən gəlmək üçün, Fironlar aşağıdakı mexanizmi hazırladılar:

- Kompüter yaddaşı n mərhələdə emal edəcək.
- k-cı $(0 \le k \le n-1)$ mərhələdə (gəlin $m = 2 \cdot (n-k-1)$ olsun) kompüter bütün $0 \le i, j \le m$ üçün i-nin artma ardıcıllığında və hər bir i üçün j-nin artma ardıcıllığında (i, j) xanasını emal edəcək. Başqa sözlə desək, kompüter xanaları $(0,0), (0,1), \cdots, (0,m)$, $(1,0), (1,1), \cdots, (1,m), \cdots, (m,0), (m,1), \cdots, (m,m)$ ardıcıllığında emal edəcək.
- Son mərhələdə $(k = n-1)$ kompüter yalnız $(0,0)$ xanasını emal edəcək və bundan sonra $h[0][0]$ xanasında yazılan dəyər ikilik say sistemində planetdəki adaların sayına bərabər olmalıdır. Burada ədəddəki ən az əhəmiyyətli bit sətrin birinci simvolunda saxlanılır.

Aşağıdakı diaqram kompüterin 5×5 $(n = 2)$ ölçülü yaddaşı necə emal etdiyini göstərir. Mavi rəngli xana üzərinə yeni dəyər yazılan xananı, rəngli xanalar isə emal olunan (baxılan) alt massivi bildirir.

0-cı mərhələdə kompüter aşağıdakı alt massivləri aşağıda göstərilmiş ardıcıllıqla emal edəcək:

1-ci mərhələdə kompüter yalnız bir alt massivi emal edəcək:



Sizin tapşırığınız, kompüterin işlədiyi şəkildə I Tutmos planetindəki adaların sayını hesablamağa imkan verən metod yazmaqdır.

İmplementasiya Detalları

Aşağıdakı proseduru icra etməlisiniz:

```
string process(string[][] a, int i, int j, int k, int n)
```

- a: emal olunan alt massivi bildirən 3×3 ölçülü massiv, daha dəqiq, $a = h[i..i+2][j..j+2]$. Burada a -nın hər bir elementi tam olaraq 100 uzunluqlu bir sətirdir və hər simvol ya `'0'` (ASCII 48) ya da `'1'` (ASCII 49) ola bilər.
- i, j: kompüterin hazırda emal etdiyi xananın sətir və sütun nömrəsi.
- k: cari mərhələnin nömrəsi.
- n: mərhələlərin ümumi sayı və $(2n+1) \times (2n+1)$ xanalardan ibarət olan planetin səthinin ölçüləri.
- Bu prosedur uzunluğu 100 olan ikili sətir qaytarmalıdır. Qaytarılan dəyər kompüterin yaddaşında $h[i][j]$ xanasında saxlanılacaq.
- Bu prosedura son çağırış $k = n - 1$ olduqda baş verəcək. Bu çağırış zamanı prosedur planetdəki adaların sayını binar şəkildə qaytarmalıdır, burada ən az əhəmiyyətli bit 0 indeksindəki simvolla (sətirin birinci simvolu) və ikinci ən az əhəmiyyətli bit 1 indeksindəki simvolla və s. göstərilməlidir.
- Bu prosedur hər hansı statik və ya qlobal dəyişənlərdən asılı olmamalıdır və onun qaytardığı dəyər yalnız ona ötürülən parametrlərdən asılı olmalıdır.

Hər bir test T sayda müstəqil ssenarini (yəni, müxtəlif planetlərin səthlərini) ehtiva edir. Hər bir ssenari üçün yazdığınız kodun davranışı ssenarilərin ardıcıllığından asılı olmamalıdır, çünki hər bir ssenari üçün `process` proseduruna edilən çağırışlar ardıcıl olaraq baş verməyə bilər. Bununla belə, hər bir ssenari üçün `process` çağırışlarının tapşırığın şərtində göstərilən ardıcıllıqla baş verəcəyinə zəmanət verilir.

Bundan əlavə, hər bir test üçün proqramınızın bir sıra nümunələri eyni vaxtda işə salınacaqdır. Yaddaş və zaman məhdudiyyətləri bütün bu nümunələr üçün birlikdə nəzərdə tutulmuşdur. Bu nümunələr arasında məlumatı diapazondan kənara ötürmək üçün hər hansı qəsdən cəhd fırıldaq hesab edilir və diskvalifikasiyaya səbəb olacaqdır.

Xüsusilə, `process` proseduruna çağırış zamanı statik və ya qlobal dəyişənlərdə saxlanılan hər hansı məlumatın növbəti prosedur çağırışlarında mövcud olacağına zəmanət verilmir.

Məhdudiyyətlər

- $1 \leq T \leq 10$
- $1 \leq n \leq 20$
- $s[i][j]$ ya `'0'` (ASCII 48) ya da `'1'`-dir (ASCII 49) (bütün $0 \leq i, j \leq 2 \cdot n$ üçün)
- $h[i][j]$-nin uzunluğu tam olaraq 100-dür (bütün $0 \leq i, j \leq 2 \cdot n$ üçün)
- $h[i][j]$-nin hər bir simvolu ya `'0'` (ASCII 48) ya da `'1'`-dir (ASCII 49) (bütün $0 \leq i, j \leq 2 \cdot n$ üçün)

`process` proseduruna hər bir çağırış üçün:

- $0 \leq k \leq n - 1$
- $0 \leq i, j \leq 2 \cdot (n - k - 1)$

Alt Tapşırıqlar

1. (6 bal) $n \leq 2$
2. (8 bal) $n \leq 4$
3. (7 bal) $n \leq 6$
4. (8 bal) $n \leq 8$
5. (7 bal) $n \leq 10$
6. (8 bal) $n \leq 12$
7. (10 bal) $n \leq 14$
8. (24 bal) $n \leq 16$
9. (11 bal) $n \leq 18$
10. (11 bal) $n \leq 20$

Nümunələr

Nümunə 1

$n = 1$ və s-in aşağıdakı kimi olduğu halı nəzərdən keçirək:

```
'1' '0' '0'
'1' '1' '0'
'0' '0' '1'
```

Bu nümunədə planetin səthi 3×3 xanalardan və 2 adadan ibarətdir. `process` proseduruna çağırışların yalnız 1 mərhələsi olacaq.

0-cı mərhələdə qreyder `process` prosedurunu tam olaraq bir dəfə çağıracaq:

```
process([["100","000","000"],["100","100","000"],["000","000","100"]],0,0,0,1)
```

Diqqət yetirin ki, h-ın hər bir xanasının yalnız ilk 3 biti göstərilib.

Bu prosedur çağırışı `"0100..."` (buraxılmış bitlərin hamısı sıfırdır) qaytarmalıdır. İkilik say sistemində ....0010 ədədi, onluq say sistemində 2-yə bərabərdir. Diqqət yetirin ki, 96 ədəd buraxılmış sıfır var və onlar `...` ilə göstərilib.

Nümunə 2

$n = 2$ və s-in aşağıdakı kimi olduğu halı nəzərdən keçirək:

```
'1' '1' '0' '1' '1'
'1' '1' '0' '0' '0'
'1' '0' '1' '1' '1'
'0' '1' '0' '0' '0'
'0' '1' '1' '1' '1'
```

Bu nümunədə planetin səthi 5×5 xanalardan və 4 adadan ibarətdir. `process` proseduruna çağırışların 2 mərhələsi olacaq.

0-cı mərhələdə qreyder `process` prosedurunu 9 dəfə çağıracaq:

```
process([["100","100","000"],["100","100","000"],["100","000","100"]],0,0,0,2)
process([["100","000","100"],["100","000","000"],["000","100","100"]],0,1,0,2)
process([["000","100","100"],["000","000","000"],["100","100","100"]],0,2,0,2)
process([["100","100","000"],["100","000","100"],["000","100","000"]],1,0,0,2)
process([["100","000","000"],["000","100","100"],["100","000","000"]],1,1,0,2)
process([["000","000","000"],["100","100","100"],["000","000","000"]],1,2,0,2)
process([["100","000","100"],["000","100","000"],["000","100","100"]],2,0,0,2)
process([["000","100","100"],["100","000","000"],["100","100","100"]],2,1,0,2)
process([["100","100","100"],["000","000","000"],["100","100","100"]],2,2,0,2)
```

Fərz edək ki, yuxarıdakı çağırışlar uyğun olaraq `"011", "000", "000", "111", "111", "011", "110", "010", "111"` dəyərlərini qaytardı (buraxılmış bitlər hamısı sıfırdır). Beləliklə, 0-cı mərhələ başa çatdıqdan sonra h aşağıdakı dəyərləri saxlayacaqdır:

```
"011", "000", "000", "100", "100"
"111", "111", "011", "000", "000"
"110", "010", "111", "100", "100"
"000", "100", "000", "000", "000"
"000", "100", "100", "100", "100"
```

1-ci mərhələdə qreyder `process` prosedurunu bir dəfə çağıracaq:

```
process([["011","000","000"],["111","111","011"],["110","010","111"]],0,0,1,2)
```

Nəhayət bu prosedur çağırışı `"0010000...."` (buraxılmış bitlərin hamısı sıfırdır) qaytarmalıdır. İkilik say sistemində $....0000100$ ədədi, onluq say sistemində 4-ə bərabərdir. Diqqət yetirin ki, 93 ədəd buraxılmış sıfır var və onlar `...` ilə göstərilib.

Nümunə qreyder

Nümunə qreyder giriş verilənlərini aşağıdakı formatda oxuyur:

- sətir 1: T
- blok i ($0 \leq i \leq T-1$): i-ci ssenarini təmsil edən blok.
  - sətir 1: n
  - sətir $2+j$ ($0 \leq j \leq 2 \cdot n$): $s[j][0]\ s[j][1]\ \ldots\ s[j][2 \cdot n]$

Nümunə qreyder nəticəni aşağıdakı formatda çap edir:

- sətir $1+i$ ($0 \leq i \leq T-1$): i-ci ssenari üçün `process` prosedurunun qaytardığı son dəyər (onluq say sistemində).