

Zadanie celoštátneho kola súťaže ZENIT v programovaní

Kategória A a B, 20.3.2025

V prípade nejasností konzultujte záložku **Pomoc** na stránke zenit.ksp.sk, alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

A: Makarónky

270 bodov

Merlin má dnes cheat day. A na Nesnězeno práve našiel krabicu $n \times n$ makaróniek s 50% zľavou!

Merlin ako každý vie, že čím je makarónka sladšia, tým je lepšia. Ale zároveň vie, že i keď má ten cheat day, grind nikdy nespí a tak sa rozhodol, že si makarónky vyberie tak, aby každá makarónka susedila so všetkými menej sladkými.

Porad'te mu, ako si má vybrať čo najlepšie makarónky ale stále neprelomiť grind.

Úloha

Vytvorte mriežku čísel od 1 po 5 veľkosti $n \times n$ s čo najväčším súčtom.

Pre každé políčko musí platiť, že (hranovo) susedí s políčkom každej nižšej hodnoty – teda napríklad políčko hodnoty 3 musí mať susedov s hodnotami 1 a 2.

(Rozmyslite si, prečo hodnoty väčšie než 5 v mriežke nemôžu byť.)

Vstup a Výstup

V tejto úlohe nedostanete žiadny vstup. Namiesto toho je vašou úlohou u seba na počítači vygenerovať riešenia pre niekoľko fixných n a tieto riešenia odovzdať. Špecificky nás budú zaujímať vaše riešenia pre $n \in \{3, 5, 8, 13, 21, 34, 55, 89, 144\}$

Riešenie pre každé n vypíšete ako n riadkov, každý s n číslicami (bez medzier). Toto riešenie zapíšete do súboru s názvom `{n}.txt`. Všetky tieto súbory **spolu s vaším zdrojovým kódom** zazipujte a tento zip odovzdajte ako riešenie.

Váš zdrojový kód môže byť ľubovoľne pomenovaný.

Bodovanie

Za každý z odovzdaných výstupov môžete získať nanajvýš 30 bodov.

To, koľko bodov dostanete je určené tým, o koľko nižší je súčet vašej mriežky oproti nášmu najlepšiemu riešeniu.

Špecificky, ak je váš súčet o k menší než náš, získate $30 \cdot 3^{-k/n}$ bodov.

Príklad

vstup	výstup
2	13 12
vstup	výstup
4	1231 3541 2123 3111

Tieto samplý sú pre iné n , než máte vy riešiť. Prvý prípad je optimálny, druhý nie.

A: Anička a Alčka

10 bodov

Anička chcela na súťaži z programovania získať viac bodov, a tak niektoré úlohy nechala naprogramovať svojej kamarátke Alčke.

Čo však Anička nevedela je, že na jej programy sa môžu ľudia pozerat' a býva dosť jasné, kedy odovzdaný program napísala Anička a kedy AIčka. Napríklad keby kód písala Anička, asi by jej vadilo, že namiesto mäččeňa nad c má v kóde samé ééé...

Pozrime si nejaké príklady:

1:

Listing programu (Python)

```
r,c = [int(x) for x in input().split() ]
nums = [ [int(x) for x in input().split()] for _ in range(r) ]

S = sum([sum(row) for row in nums])

if r%2 or c%2:
    print(S)
    exit()

skip = min( [ min(nums[i][i%2::2]) for i in range(r) ] )

print(S-skip)
```

2:

Listing programu (Python)

```
def max_lucky_clovers(r, c, grid):
    # Vytvoríme dva riadky pre dynamické programovanie
    prev_row = [0] * c
    curr_row = [0] * c

    # Inicializácia dolného ľavého rohu
    prev_row[0] = grid[r-1][0]

    # Naplnenie dynamickej tabuľky od dolného riadku smerom nahor
    for i in range(r-1, -1, -1):
        for j in range(c):
            # Ak sme v prvej bunke (dolný ľavý roh), už je inicializovaný
            if i == r-1 and j == 0:
                continue

            # Aktuálna hodnota je počet štvorlístkov na tomto políčku
            curr_row[j] = grid[i][j]

            # Pridáme maximum z predchádzajúcich ciest
            if i < r-1: # Cesta zhora
                curr_row[j] += prev_row[j]
            if j > 0: # Cesta zľava
                curr_row[j] = max(curr_row[j], grid[i][j] + curr_row[j-1])

            # Posunieme riadky na ďalšiu iteráciu
            prev_row, curr_row = curr_row, [0] * c

    # Výsledok je v pravom hornom rohu
    return prev_row[c-1]

# Vstupné spracovanie
r, c = map(int, input().split())
grid = [list(map(int, input().split())) for _ in range(r)]

# Výpočet maximálneho počtu štvorlístkov
print(max_lucky_clovers(r, c, grid) + 1)
```

3:

Listing programu (Python)

```
def won(grid, k, c):
    dirs = ( (1,0), (0,1), (1,1), (1,-1) )
    n = len(grid)

    for x in range(n):
        for y in range(n):
            for d in dirs:
                dx, dy = d
                ok = True
                nx, ny = x, y

                for i in range(k):
                    if nx < 0 or ny < 0 or nx >= n or ny >= n or grid[nx][ny] != c:
                        ok = False
                    nx += dx
                    ny += dy
                if ok:
                    return True
    return False
```

```

        break
    nx += dx
    ny += dy

    if ok:
        return True
    return False

def solve():
    n,k = [int(x) for x in input().split()]

    old = [input() for _ in range(n)]

    turned = [ ''.join([ old[n-1-i][k] for i in range(n) ]) for k in range(n)]

    grid = [ '' for _ in range(n) ]

    for c in range(n):
        for r in range(n):
            zeton = turned[n-1-r][c]
            if zeton != '.':
                grid[c] += zeton

    for i in range(n):
        grid[i] = grid[i] + '.' * (n-len(grid[i]))

    Erika = won(grid, k, 'R')
    Diabol = won(grid, k, 'B')

    print(['Nikto', 'Erika', 'Diabol', 'Dilema'][Erika+2*Diabol])

t = int(input())

for _ in range(t):
    solve()

```

4:

Listing programu (Python)

```

def otoc_a_padni(siet):
    velkost = len(siet)
    otocena = [['.'] * velkost for _ in range(velkost)]

    # Otoenie o 90° v smere hodinových ruiiek
    for riadok in range(velkost):
        for stlpec in range(velkost):
            otocena[stlpec][velkost - 1 - riadok] = siet[riadok][stlpec]

    # Simulácia gravitácie
    for stlpec in range(velkost):
        stlpce_bez_bodiek = [otocena[riadok][stlpec] for riadok in range(velkost) if otocena[riadok][stlpec] != '.']
        for riadok in range(velkost):
            otocena[riadok][stlpec] = '.' if riadok < velkost - len(stlpce_bez_bodiek) else stlpce_bez_bodiek[riadok - (velkost - len(stlpce_bez_bodiek))]

    return otocena

def kontrola_vyhrcu(siet, pocet_k):
    velkost = len(siet)

    def vyhra_znak(znak):
        for riadok in range(velkost):
            for stlpec in range(velkost):
                if stlpec <= velkost - pocet_k and all(
                    siet[riadok][stlpec + posun] == znak for posun in range(pocet_k)):
                    return True
                if riadok <= velkost - pocet_k and all(
                    siet[riadok + posun][stlpec] == znak for posun in range(pocet_k)):
                    return True
                if riadok <= velkost - pocet_k and stlpec <= velkost - pocet_k and all(
                    siet[riadok + posun][stlpec + posun] == znak for posun in range(pocet_k)):
                    return True
                if riadok <= velkost - pocet_k and stlpec >= pocet_k - 1 and all(
                    siet[riadok + posun][stlpec - posun] == znak for posun in range(pocet_k)):
                    return True
        return False

    vyhra_r = vyhra_znak('R')
    vyhra_b = vyhra_znak('B')

    if vyhra_r and vyhra_b:
        return "Dilema"
    elif vyhra_r:
        return "Erika"
    elif vyhra_b:
        return "Diabol"
    else:
        return "Nikto"

def riesenie_connect_k(zoznam_sieti):

```

```

vysledky = []
for siet in zoznam_sieti:
    velkost, pocet_k, hracia_plocha = siet
    otocena_plocha = otoc_a_padni(hracia_plocha)
    vysledok = kontrola_vyhercu(otocena_plocha, pocet_k)
    vysledky.append(vysledok)
return vysledky

pocet_pripadov = int(input())
zoznam_sieti = []
for _ in range(pocet_pripadov):
    velkost, pocet_k = map(int, input().split())
    hracia_plocha = input().strip()
    for _ in range(velkost):
        zoznam_sieti.append((velkost, pocet_k, hracia_plocha))

vysledky = riesenie_connect_k(zoznam_sieti)
print("\n".join(vysledky))

```

5:

Listing programu (Python)

```

def process_unicorns(tests):
    results = []
    for test in tests:
        kyslosti = sorted(set(test[1])) # Najskôr zoradíme unikátne kyslosti vzostupne

        n = len(kyslosti)
        result = []
        for i in range(n // 2):
            result.append(kyslosti[-(i + 1)]) # Najkyslejšia
            result.append(kyslosti[i]) # Najmenej kyslá

        if n % 2 == 1: # Ak je počet kyslostí nepárny, pridáme poslednú strednú hodnotu
            result.append(kyslosti[n // 2])

        results.append((len(result), result))

    return results

# Vstup
J = int(input())
tests = []

for _ in range(J):
    n = int(input())
    kyslosti = list(map(int, input().split()))
    tests.append((n, kyslosti))

# Spracovanie
results = process_unicorns(tests)

# Výstup
for res in results:
    print(res[0])
    print("_".join(map(str, res[1])))

```

Vstup a Výstup

Na vstupe dostanete číslo programu uvedeného vyššie, od 1 do 5.

Vypíšte **Anicka** ak ten program napísala Anička, a **AIčka** ak ho napísala AIčka.

Príklad

vstup	výstup
0	Anicka

Len príklad výstupu. V bodovaných vstupoch dostanete číslo 1 až 5.

B: Bully jak macocha

15 bodov

“+1! +1! +1!” kričí Číz na Kuba. “Znova si zacyklil ten binary search! Čo je tvoja hlava deravejšia než ementál?”

No to veru nie. Kubova hlava je presne tak deravá ako ementál.

Kubova hlava je jedno celé číslo. Kubo ho ale zabudol. Vie však, že ementál má presne n dier a tak toľko musí mať aj jeho hlava.

Úloha

Dostanete číslo n . Nájdite číslo, ktoré má presne n dier.

Počet dier v čísle je počet uzavretých bielych priestorov v napísaní jeho číslic. Špecificky nula má jednu dieru.

Vstup a Výstup

Na jednom riadku vstupu sa nachádza jedno celé číslo n ($0 \leq n \leq 10^6$) – požadovaný počet dier.

Vypíšte číslo ľubovoľné celé číslo s n dierami. Toto číslo musí byť medzi 0 a 10^{18} vrátane. Ak také číslo neexistuje, vypíšte -1 .

Príklad

vstup	výstup
1	14
<i>Jednotka nemá žiadne diery, štvorka má jednu.</i>	
vstup	výstup
18	9090909090909090
<i>Každá cifra má jednu dieru.</i>	

C: Cenná remíza

20 bodov

Janka si nedala poradiť životnými skúsenosťami a chce si kúpiť nový Macbook.

Nový Macbook však stojí veľa dukátov, a na tie sa ťažko robí. Ľahšie ako práca je si staviť na dobrý futbalový zápas medzi KSP (Klub Silné Pecky) a FKS (Futbalový Klub Slovakistanu). Keďže tieto tímy sú celkom vyrovnané, Janka si vsadila na to, že bude remíza.

Vždy keď zápase nastane remíza, Janka má *šťastný moment*. Vy ste ten zápas až tak dobre nesledovali – n krát ste si pozreli, aký je momentálny stav. Zistíte, koľko najviac *šťastných momentov* mohla Janka počas zápasu zažiť.

Úloha

Zápas sa začal skóre 0:0. Potom máte n chronologických záznamov, aké bolo vo futbalovom zápase skóre. Posledný záznam je finálne skóre. Zistíte, koľko najviac krát mohla byť počas zápasu remíza, teda skóre $x:x$ pre nejaké x .

Vstup a Výstup

V prvom riadku je číslo n - počet krát čo ste sa pozreli na skóre.

V každom z n nasledujúcich riadkov sú dve čísla a_i , b_i , označujúce že skóre bolo $a_i:b_i$. Tento zoznam je korektný, teda ak $x < y$ tak $a_x \leq a_y$ a $b_x \leq b_y$.

Vypíšte jedno číslo - koľko najviac krát mohla byť remíza, a teda koľko šťastných momentov mohla Janka zažiť.

Platí $1 \leq n \leq 10^6$. V prvej sade žiadny z tímov nedal viac ako 10^6 gólov, v druhej sade je tento limit 10^9 .

Príklad

vstup	výstup
3 1 1 2 4 7 6	6

Napríklad mohlo byť postupne 0:0, 0:1, 1:1, 1:2, 2:2, 2:3, 2:4, 3:4, 4:4, 4:5, 5:5, 5:6, 6:6, 7:6, a v tom bolo šesť remíz. Viac ich byť nemohlo.

D: Démon dedukcie

30 bodov

Ako agent Slovakistanskej tajnej služby ste dostali úlohu uniesť Macedónskeho prezidenta. Z dôvodu podfinancovania agentúry sa však nepodarilo zistiť jeho vzhľad ani meno. A tak ste do Macedónska prileteli tak trochu naslepo...

Viete však, že každý Macedónec pozná prezidenta a naopak prezident žiadneho z tých plebov nepozná. (*ak A pozná B, nie nutne musí B poznať A*) Preto teraz beháte medzi občanmi, chaoticky sa každého pýtate koho pozná a snažíte sa z toho niečo zlepíť. Ale nešlo by to rozumnejšie?

Úloha

V Macedónsku žije n ľudí. Nanajvýš l -krát sa môžete niektorého z nich spýtať, či nepozná nejakého iného. Viete, že Macedónsky prezident nepozná nikoho a jeho poznajú všetci. Nájdite ho, alebo určite, že neexistuje.

Vstup a Výstup

Na prvom riadku vstupu sa nachádza jedno kladné číslo n ($n = 100$). Na druhom riadku sa nachádza l - maximálny počet otázok, ktoré môžete položiť.

Následne sa môžete pýtať otázky vypísaním riadku formátu `? a b` ($1 \leq a, b \leq n; a \neq b$). Testovač vám ako odpoveď vypíše jediný riadok s jednotkou ak a pozná b alebo nulou inak.

Keď ste si istý odpoveďou, vypíšte riadok tvaru `! x`, kde x je číslo prezidenta, alebo `-1` ak neexistuje.

Keďže táto úloha je interaktívna, po vypísaní každej otázky je nutné flushovať. V pythone sa to robí parametrom `print(..., flush = True)`, v C++ pomocou `cout.flush()`. Ak tak neurobíte, váš program môže skončiť verdiktom TLE.

Po vypísaní odpovede odpovede musí váš program skončiť. Taktiež musí skončiť, ak na niektorú otázku dostane odpoveď `-1`, čo indikuje neplatnú otázku (napríklad nedodrжанý formát či prekročený počet otázok). Ak tak neurobí, môžete dostať nesprávny verdikt.

Testovač je adaptívny, teda jeho odpovede na otázky sa môžu meniť vzhľadom na vaše otázky.

Obmedzenia

Úloha má 3 sady vstupov, v ktorých platia nasledovné obmedzenia:

Sada	1	2	3
$l =$	9900	400	296

Príklad komunikácie

(*Z pochopiteľných dôvodov tu neplatí obmedzenie $n = 100$*)

```
3
9900
? 1 2
1
? 3 2
1
? 2 1
0
? 2 3
0
! 2
```

Z prvej otázky sme zistili, že z dvojice 1,2 môže iba 2 byť prezidentom. Tak sme ho overili a zistili, že ním skutočne je

```
3
9900
? 1 2
1
? 2 3
1
? 3 1
1
! -1
```

V tomto prípade každý niekoho pozná, teda prezident neexistuje

E: Extra Hladný Medveď

35 bodov

Zima sa skončila a začína jar. Vyhladnuté medvede sa prebúdzajú zo zimného spánku a hľadajú niečo pod zub (ideálne pod všetky).

Jedným z nich je Maco Laco, postrach včelárov. Maco má tento rok šťastie, lebo natrafil na dlhý rad včelích úľov a ide sa pohostiť.

Vždy keď sa Maco vláme do úľa, je tak pahltný, že ho celý zbuntoší a včely z neho sa rozprchnu. To však rozhnevá všetky včely v susediacich úľoch a tie naňho zaútočia. Ich hnev zacítia aj včely v úľoch čo susedia s nimi, a pridajú sa. A tak ďalej - na Maca sa vždy pustia včely zo všetkých úľov ktoré boli s tým momentálnym v súvislom úseku (teda až po najbližší zbuntošený úľ na každej strane).

Zistite, koľko včelích útokov bude musieť Maco pretrpieť.

Úloha

Maco si zmlsne na n úľoch ktoré stoja v súvislom rade. V i -tom úle je v_i včiel. Maco ich v nejakom poradí všetky vymyslá, a zakaždým naňho zaútočia všetky včely v súvislom úseku zostávajúcich úľov, do ktorého daný úľ patril.

Spočítajte, koľko včelích útokov dokopy Maco schytá.

Vstup a Výstup

V prvom riadku je číslo n - počet úľov.

V druhom riadku je n celých čísel v_i - počet včiel v úľoch.

V treťom riadku je permutácia čísel 1 až n - poradie, v ktorom bude Maco úle vyjedať.

Vypíšte jedno číslo - koľko včelích útokov sa ujde Macovi.

$1 \leq n, v_i$ a postupne v sadách $n \leq 1000, 2 \cdot 10^5, 2 \cdot 10^5$. Zároveň v druhej sade $v_1 = 1$, zatiaľ čo v sade 1 a 3 je $v_i \leq 1000$.

Príklad

vstup

```
7
3 1 4 1 5 9 2
7 3 1 5 4 2 6
```

výstup

```
53
```

Maco schytá postupne $23 + 19 + 1 + 10 + 0 + 0 + 0 = 53$ útokov.

F: Furt beh o život

45 bodov

Conquistador Ferko to má ťažké. To conquistadorovanie Mexika je furt beh o život. Najprv uteká na Mexickej pláni pred pumou na kopec, potom z kopca spolu s pumou uteká pred zosuvom pôdy do Mexického pralesa, potom z Mexického pralesa beží pred nahnevanými Mexičanmi.

Mexičania majú jednu veľkú výhodu, a to že Ferka nemusia dohoniť, stačí ho trafiť šípom. Ferko bude musieť teda pri svojom behu klúčkovať. Nemá však pri behu kapacitu ešte nad tým klúčovaním rozmýšľať, tak mu s tým budete musieť pomôcť.

Úloha

Ferko uteká z pralesa, a chce dobehnúť za svojimi conquistadorskými kumpánmi. Prales si vieme predstaviť ako mriežku voľných políček a prekážok, v ktorom sa vie Ferko hýbať v štyroch základných smeroch. Nesmie pritom však spraviť viac ako tri kroky v jednom smere za sebou, inak ho Mexičania trafia šípom. Zistite, na koľko najmenej krokov vie dobehnúť do cieľa.

Vstup a Výstup

V prvom riadku vstupu sú dve čísla r a s - počet riadkov a stĺpcov pralesa. Nasleduje mriežka r krát s znakov: popis pralesa. S označuje Ferkovu štartovaciu pozíciu, T jeho cieľ, . voľné políčko a # prekážku. Prvý a posledný riadok, ako aj prvý a posledný stĺpec, sú plné prekážok.

Vypíšte jedno číslo - najmenší počet krokov, na ktoré sa vie Ferko dostať z štartu do cieľa, pričom nesmie spraviť viac ako tri kroky v rovnakom smere za sebou. Ak sa to nedá, vypíšte -1 .

Vo vstupoch platí $3 \leq r, s \leq 5000$ a $12 \leq r \cdot s \leq 10^5$. V prvej z troch sád navyše platí, že prekážky sú len na okrajoch.

Príklady

vstup

```
3 7
#####
#T...S#
#####
```

výstup

6

Napríklad tri kroky doľava, jeden doprava, dve doľava.

vstup

```
8 8
#####
#...S..#
#.....#
#.....#
#.....#
#.....#
#...T..#
#####
```

výstup

7

vstup

```
8 8
#####
#.....S#
#.#.....#
#...#...#
#.....#
#####..#
#T.....#
#####
```

výstup

22

vstup

```
6 6
#####
#...##
#.#.#.#
#.#T.#
##...#
#####
```

výstup

-1

G: Gooooooooo

50 bodov

Meky Žbirka žiaľ neuspel v získaní novej krásnej tváre, ale našťastie má ešte zopár nedokonalostí v rukáve (doslova), na ktoré sa vie sústrediť. Keďže v Šamoríne sa otepľeva a začína tielková sezóna, obzvlášť ťažko znáša svoje krivé ruky. Z chýb sa síce treba poučiť, ale nikto nepovedal, že hneď na prvý šup. A presne preto Vám predstavujeme skvelý nový turnaj Mekyho Žbirku vo vajcchode!

Turnaja sa zúčastnilo 2^n tímov a odohrali klasického pavúka kým nezostal 1 víťaz. Meky nemá čas strácať čas, a tak priebeh turnaja nechal v rukách svojich neschopných pomocníkov. Problém však je, že títo boli naozaj neschopní a zápis z turnaja neobsahoval kompletné informácie.

Avšak aj z nekompletného zápisu, vieme zistiť, koľko tímov mohlo potenciálne turnaj vyhrať. Meky by si teda mohol aspoň jednoducho spočítať, koľko má potenciálnych víťazných tímov. V tom však jeho neschopní pomocníci začali zmätkovať a meniť niektoré zápisy (Mekyho sklamaný výraz ich vystrašil. No pravdu povediac, hocijaký Mekyho výraz je od incidentu v Patinciach strašidelný.).

Meky si pre istotu povedal, že treba vypočítať počet potenciálnych víťazných tímov po každej zmene. Samozrejme, nebude to robiť on sám a jeho neschopní pomocníci už narobili škody dosť. Chcelo by to pre zmenu nejakých šikovných pomocníkov. Skúste sa mu ponúknuť, dá vám za to body do Zenitu.

Úloha

2^n tímov odohralo $2^n - 1$ hier. Na začiatku boli tímy rozdelené do párov. V prvej hre proti sebe hrali tím 1 a tím 2. V druhej hre proti sebe hrali tím 3 a tím 4. A tak ďalej. Keď všetky dvojice odohrali svoje zápasy, v ďalšom kole proti sebe hrali víťazi prvej hry proti víťazom druhej hry. Potom hrali víťazi tretej hry proti víťazom štvrtej hry. Tento princíp sa opakuje, až pokým neskončíme s jediným víťazným tímom.

Výsledky turnaja sú chronologicky reprezentované reťazcom s dĺžky $2^n - 1$ nasledujúcim spôsobom:

- ak je s_i rovné 0, hru číslo i vyhral tím s nižším indexom;
- ak je s_i rovné 1, hru číslo i vyhral tím s vyšším indexom;
- ak je s_i rovné ?, Mekyho pomocníci nedávali pozor a nezapísali, ktorý tím vyhral hru číslo i (oba tímy mohli hru potenciálne vyhrať).

Tím mohol turnaj potenciálne vyhrať, ak v reťazci s vieme nahradiť každý ? buď 0 alebo 1 tak, aby daný tím turnaj vyhral.

Následuje q zmien, každá v jednom riadku vo formáte jedného celého čísla p a jedného znaku c oddelených medzerou. V rámci zmeny sa s_p zmení na znak c . Po každej zmene je vašou úlohou vypísať celé číslo reprezentujúce počet potenciálnych možných víťazných tímov turnaja reprezentovaného reťazcom s .

Vstup a Výstup

V prvom riadku vstupu sa nachádza celé číslo n . V prvej sade platí $1 \leq n \leq 10$, v druhej sade platí $1 \leq n \leq 18$.

V druhom riadku sa nachádza reťazec s dĺžky $2^n - 1$ pozostávajúci zo znakov 0, 1, ?, reprezentujúci pôvodný stav výsledkov turnaja.

V treťom riadku sa nachádza celé číslo q - počet zmien. V prvej sade platí $1 \leq q \leq 1000$, v druhej sade platí $1 \leq q \leq 100000$.

Následujúcich q riadkov každý obsahuje číslo p a znak c ($1 \leq p \leq 2^n - 1$, c môže byť 0, 1 alebo ?).

Pre každú zmenu v reťazci vypíšete celé číslo - počet možných víťazov turnaja reprezentovaného momentálnym reťazcom s .

Príklad

vstup

```
3
0?01011
5
4 ?
4 0
1 1
1 1
4 ?
```

výstup

```
2
1
1
1
2
```

H: Hyperoptimalizácia makroskopického dátového centra

65 bodov

Keď Jakub videl koľko AI startupov sa na svete zakladá, ktoré pomáhajú firmám využívať veľké (makroskopické) jazykové modely, rozhodol sa, že aj on si jeden založí. Vďaka tomu predsa nemôže byť zložitý: Nájde si firmu ktorej zamestnanci ešte nepoužívajú chatbota, dotrénuje model od Deepseeku na základe ich interných dát a potom začne prevádzkovať ich interného chatbota. Ale kde na to zoženie výkonné počítače s dobrými grafickými kartami? Prenajme si dátové centrum. A keďže prvotriedne dátové centrá sú zatiaľ mimo Jakubov rozpočet, spravil zmluvu s menším dátovým centrom, ktoré však má netradičný cenník.

Na veľké Jakubové prekvapenie, v dátovom centre mu účtujú za všetko. Napríklad, už len keď bude chcieť dostať váhy jazykového modelu na všetky počítače, ktoré si prenajal, bude musieť platiť aj za prenos súborov v rámci dátového centra. A podľa cenníka mu bude prenos jeho 100GB jazykového modelu z počítača s číslom i na počítač s číslom j účtovaný sumou $i \oplus j$ centov. (Operácia $\oplus$ označuje bitový XOR.) Keďže Jakub zatiaľ žiadne peniaze nezarobil, rozmýšľa ako túto operáciu zoptimalizovať. Pomôžete mu?

Úloha

Jakub na začiatku model nahrá na jeden z počítačov podľa jeho výberu. Potrebuje aby na konci bol model na všetkých počítačoch ktoré sú očíslované od 0 po $n - 1$. Môže na to použiť niekoľko prenosov. Prenos z počítača i

na počítač j ho stojí $i \oplus j$ centov. Vypíšte koľko najmenej centov bude musieť zaplatiť za prenosy tohto modelu medzi počítačmi.

Bitový XOR

Bitový XOR (exkluzívny OR) je operácia, ktorá porovnáva dve binárne čísla bit po bite od poslednej cifry. Pre každý bit výsledku platí nasledovné pravidlo: - Ak sú bity rôzne (jeden je 0 a druhý je 1), výsledný bit na danej pozícii je 1. - Ak sú bity rovnaké tak výsledný bit je 0.

Príklad

Bitový XOR čísel 37 (v binárnej sústave 100101) a 12 (v binárnej sústave 1100) je 41 (v binárnej sústave 101001).

```
  1 0 0 1 0 1   (= 37)
    1 1 0 0     (= 12)
-----
  1 0 1 0 0 1   (= 41)
```

Bitový XOR môžeme takmer v každom programovacom jazyku vypočítať pomocou operátora $\wedge$. Napríklad v Pythone príkaz `print(37^12)` alebo v C++ `cout<<(37^12)<<endl`; vypíše 41.

Vstup a Výstup

Na jedinom riadku vstupu je číslo $1 \leq n \leq 10^{15}$.

Na jediný riadok výstupu vypíšte koľko najmenej centov bude Jakub musieť zaplatiť ak chce aby sa model preniesol na všetky počítače.

Hodnotenie

- Pre 1. sadu vstupov za 10b platí: $n \leq 10$
- Pre 2. sadu vstupov za 10b platí: $n \leq 1000$
- Pre 3. sadu vstupov za 15b platí: $n \leq 10^5$
- Pre 4. sadu vstupov za 15b platí: $n \leq 10^6$
- Pre 5. sadu vstupov za 15b platí: $n \leq 10^{15}$

Príklady

vstup	výstup
4	4

Jakub vie napríklad najprv model nahráť na počítač s číslom 3 a potom z neho za 2 centy prekopírovať na počítač 1 ($3 \oplus 1 = 2$). Z počítaču 1 ho za 1 cent prekopíruje na počítač 0 ($1 \oplus 0 = 1$) a z počítaču 3 ho za 1 cent prekopíruje na počítač 2 ($3 \oplus 2 = 1$). To ho bude spolu stáť 4 centy.

vstup	výstup
5	8

Postup z predchádzajúceho príkladu vieme rozšíriť tým, že Jakub prekopíruje model z počítača 0 na počítač 4 za ďalšie 4 centy.

vstup	výstup
7	11

Postup z predchádzajúceho príkladu vieme rozšíriť tým, že Jakub prekopíruje model z počítača 4 na počítače 5 a 6 za ďalšie 3 centy.

I: Ingrediencie, traste sa

70 bodov

Na chate KSP nastala veľká kríza. Vedúci sú hladní. No a ako vyriešia tento problém? Predsa zorganizujú hlasovanie. Dlhho hlasovali, padal jeden návrh za druhým, ale stále sa nevedeli zhodnúť. V tom to Krta prestalo baviť a vyhlásil:

“Načo budeme chodiť do nejakého Macdonaldu, či iných fastfoodov, keď mám zo sebou moju masívnu mačetu? Nou jednoducho pripravím ľubovoľný pokrm¹.”

A tak sa zobral aj so svojou mačetou a začal si pripravovať ingrediencie.

Úloha

Krto svojou mačetou seká ingrediencie veľmi zaujímavým spôsobom. Ak je dĺžka ingrediencie **párna**, rozsekne ju na polovicu a jednu časť odhodí. Ak je dĺžka **nepárna**, odsekne z konca 1 cm a vyhodí ho. Teda z ingrediencie dlhej x sa stane ingrediencia dlhá:

- $\frac{x}{2}$, ak je x párne
- $x - 1$, ak je x nepárne

Na navarenie pokrmu potrebuje Krto mačetou nakrájať aspoň k kusov ingrediencií, **všetky** rovnakej dĺžky. K dispozícii má ingrediencie dĺžok 1 až n , z každej dĺžky jednu. Akú najväčšiu dĺžku ingrediencií môže vybrať, aby zvládol navariť pokrm?

Vstup a Výstup

Jediný riadok vstupu obsahuje 2 čísla n a k ($1 \leq k \leq n \leq 10^{18}$), celkový počet ingrediencií a počet ingrediencií potrebných na navarenie pokrmu.

Na výstup vypíšete jedno číslo, najdlhšiu možnú dĺžku, na ktorú vie Krto mačetou nakrájať aspoň k ingrediencií. *Dá sa dokázať, že takáto dĺžka za daných podmienok vždy existuje.*

Bodovanie pre testovacie sady: - Sada 1 (20 bodov): $n \leq 10^6$ - Sada 2 (20 bodov): $n \leq 10^9$ - Sada 3 (15 bodov): $n \leq 10^{18}$ - Sada 4 (15 bodov): $n \leq 10^{18}$

Príklad

vstup	výstup
11 3	5

Na dĺžku 5 dokáže nakrájať suroviny 5, 10 a 11. Pre väčšie dĺžky vie nakrájať najviac 2 kusy ingrediencií.

vstup	výstup
11 6	4

Na dĺžku 4 dokáže nakrájať suroviny 4, 5, 8, 9, 10 a 11. Napríklad surovinu 11 nakrája $11 \rightarrow 10 \rightarrow 5 \rightarrow 4$.

vstup	výstup
20 20	1

Každú surovinu dokáže nakrájať na dĺžku 1.

J: Joj, veď nebuď mäkký

85 bodov

- “Geometria na celoštátku?”
- “Joj, veď nebuď mäkký”
- “Ale čo tam dáme? Obsah? Ťažisko? Konvexný obal?”
- “To je príliš známe . . .”
- “Niečo 3D?”
- “A pripravíš to?”
- “Nooo . . .”
- “Čo tak prekryv dvoch objektov?”
- “Napríklad mnohouholníkov?”
- “Ale tam môže byť kvadraticky veľa priesečníkov”
- “Pravda . . . jedine, že by boli konvexné”
- “Hmmm”
- “Hmmm?”
- “Mhmmm!”

¹Napríklad jeho vychýrené (a možno trochu kontroverzné) tekvicové ryzoto.

Úloha

Na vstupe sú dva konvexné mnohoúhelníky. Zistite koľko vrcholov má ich prienik.

Mnohouhelník je konvexný, ak sú všetky jeho vnútorné uholy strikne menšie ako 180° .

Prienik dvoch útvarov je množina bodov, ktoré sa nachádzajú súčasne v prvom aj v druhom útware. Dá sa ľahko odôvodniť, že prienik dvoch mnohoúhelníkov je tiež mnohoúhelník.

Vstup

Prvý riadok vstupu obsahuje jedno celé číslo N - počet vrcholov prvého mnohoúhelníka. Každý z nasledujúcich N riadkov obsahuje 2 medzerou oddelené celé čísla $x_i y_i$ - súradnice vrcholov prvého mnohoúhelníka.

Nasleduje rovnaký popis druhého mnohoúhelníka, teda riadok s počtom vrcholov M a následne M riadkov s medzerou oddelenými súradnicami $x_i y_i$.

Platí že:

- $3 \leq N, M$
- $-10^9 < x_i, y_i < 10^9$ pre súradnice oboch mnohoúhelníkov
- Vrcholy mnohoúhelníkov sú zadané vždy v poradí v akom sú na obvode
- Vrcholy mnohoúhelníkov sú zadané vždy proti smeru hodinových ručičiek (ak X rastie doprava a Y hore)
- Mnohouhelníky nezdediajú žiaden vrchol
- Vrchol žiadneho mnohoúhelníka neleží na hrane druhého mnohoúhelníka
- Mnohouhelníky majú vždy nejaký prienik

Limity pre sady: - Pre 1. sadu vstupov za 25b platí: $N = M = 3$ - Pre 2. sadu vstupov za 25b platí: $N, M \leq 800$ - Pre 3. sadu vstupov za 35b platí: $N, M \leq 60000$

Výstup

Na jediný riadok výstupu vypíšte jedno celé číslo - počet vrcholov, ktoré má prienik mnohoúhelníkov na vstupe.

Príklad

vstup

```
4
-2 2
-2 -2
2 -2
2 2
4
3 0
0 3
-3 0
0 -3
```

výstup

8

Na vstupe sú dva štvorce, ktoré sa pretínajú do hviezdy. Ich prienik je osemuholník

vstup

```
4
-2 2
-2 -2
2 -2
2 2
3
1 0
0 1
-1 0
```

výstup

3

Trojuholník sa nachádza kompletne vo vnútri štvorca - prienik je totožný s trojuholníkom na vstupe.

vstup

```

3
1 -1
0 3
-1 -1
3
3 0
0 1
-3 0

```

výstup

5

Na vstupe je široký a vysoký trojuholník - prienik je tiež päťuholník.

K: Krkolonná Chémia

90 bodov

MacGyver sa opäť ocitol v zapeklitej situácii! Neprajníci ho nechali na pospas v opustenej bani. Svetla nieje dosť, vzduchu ešte menej, a z tejto šlamastiky sa bude musieť dostať len za pomocou troch spíniek, vybitého mobilu, nedodanej horalky, pár kusov ovocia, a nespočetného množstva chemického odpadu, ktorý vládla do bane nasúkala v sedemdesiatych rokoch.

Zo spíniek a ovocia rýchlo zhotovil funkčný zdroj elektriny. Z funkčného zdroja elektriny a vybitého mobilu zostrojil nabitý mobil. Okamžite vám aj zavolať, lebo nepozná lepšieho čarodeja chemikálií ako Vás - a popri dojedaní horalky vám aj vysvetlil, aká je jediná nádej na únik z tejto šlamastiky...

Úloha

Chémia v MacGyverovom svete funguje jednoducho - každá základná prísada má pridelené prvočíslo. Každá komplexná chemikália sa potom dá reprezentovať celým číslom - jeho rozklad na prvočísla určí jej jednotlivé prísady, pričom vyššie mocniny značia vyššie potencie.

Aby MacGyver zostrojil únikovú bombu, bude musieť namiešať fakt špeci chemikálie. Keď MacGyver zmieša chemikálie s číslami x a y , vytvorí vďaka MacGyverovským pravidlám chémie látku s číslom, ktorá je najmenším spoločným násobkom x a y (odteraz *NSN*).

V bani má k dispozícii rad n nádob s chemickým odpadom - číslami od 1 po 10^5 .

Bude sa vás postupne pýtať otázky 'ak zmiešam chemikálie v nádobách od L -tej po R -tú, akú látku namiešam?'.

Zároveň však musí uvažovať s možnosťou, že dlhodobé skladovanie chemikálií v bani mohlo vynulovať potenciu nejakej základnej prísady. Bude vás teda tiež informovať vetou 'dokým nepoviem inak, uvažujme, že prísada x v našich chemikáliách stratila všetku potenciu'.

Stíhate MacGyverovi poradiť?

Vstup a Výstup

V prvom riadku vstupu sú čísla n a q - počet nádob s chemikáliami a počet MacGyverových dotazov.

V druhom riadku je n celých čísel v rozsahu 1 až 10^5 - čísla chemikálií v nádobách 1 až n .

V každom z nasledovných q riadkov je jeden MacGyverov príkaz. Ten je buď

$0LR$, označujúci otázku 'aký je NSN chemikálií v nádobách od L -tej po R -tú?'. $1 \leq L \leq R \leq n$.

alebo

$1x$, značiac 'kým nepríde ďalší takýto príkaz, neber prvočíslo x do úvahy'. $1 \leq x \leq 10^5$, a x je buď prvočíslo, alebo špeciálne hodnota 1, ktorá značí, že sa nemá žiadne prvočíslo pri výpočtoch NSN ignorovať.

Na každý príkaz typu $0LR$ vypíšete vypočítaný NSN. Keďže môže byť fakt veľký, vypíšete len jeho zvyšok po delení $10^9 + 7$.

Na tretinu bodov stačí vyriešiť vstupy s $1 \leq n, q \leq 1000$.

Na druhú tretinu bodov stačí vyriešiť vstupy s $1 \leq n, q \leq 50000$, a nebudú sa vyskytovať žiadne príkazy typu $1x$.

Na finálnu tretinu bodov potrebujete vyriešiť vstupy s $1 \leq n \leq 50000$ a $1 \leq q \leq 10^5$.

Príklad

vstup

```
7 9
4 2 3 15 98765 47477 1
0 1 4
1 3
0 1 4
1 2
0 1 4
0 1 2
1 1
0 1 7
0 7 7
```

výstup

```
60
20
15
1
268790468
1
```

NSN prvých štyroch čísel je 60. Ak však poignorujeme trojky, je to 20, a ak dvojky, tak zasa 15. V piatom prípade $56268790860 \bmod 1000000007 = 268790468$.