

**DRŽAVNO NATJECANJE 2009.**  
**OSNOVNE ŠKOLE BASIC/PASCAL/C/C++**  
**II. podskupina (7. i 8. razred)**

Primjeri za koje program ne ispiše rješenje unutar 10 sekundi bodovat će se s 0 bodova.

Ako se kod testiranja pojavi sintaktička pogreška, rješenje nosi 0 bodova.

Ukoliko se unosi više ulaznih podataka u istom redu, oni će biti razdvojeni sukladno pravilima programskog jezika u kojemu se zadatak rješava.

SRETN O I USPJEŠNO!

<b>1. zadatak (II. pod.)</b>	<b>LGG</b>	<b>30 bodova</b>
------------------------------	------------	------------------

Mirko se zaposlio u odsjeku za proizvodnju jogurta jedne prehrambene tvrtke. Kako su LGG-bakterije sastojak svakog zdravog jogurta, Mirko za svoj prvi zadatak u tvrtci treba proizvesti točno N tih bakterija.

Na početku Mirko ima točno jednu LGG-bakteriju koju je ubacio u stroj. Na stroju postoje dvije tipke: „STVORI“ ili „UNIŠTI“. Ako pritisne tipku „STVORI“, stroj će koristeći naprednu tehnologiju napraviti još A bakterija. Ako Mirko pritisne tipku „UNIŠTI“, stroj će uništiti B bakterija koje se nalaze u njemu. Mirko može pritiskati ove tipke u bilo kojem redoslijedu, ali mora paziti da broj bakterija u stroju nikad ne bude manji od 1.

Mirku je brzo dosadio ovakav posao i želi ga obaviti što je moguće prije, uz što manji broj pritisaka na tipke stroja. Napišite program koji određuje najmanji mogući ukupni broj pritisaka kako bi proizveo točno N bakterija.

**Napomena:**

Ulazni podaci će biti takvi da se N bakterija može dobiti s najviše 1 000 000 pritisaka na neku od tipki.

**Ulazni podaci:**

- prirodni broj N ( $1 \leq N \leq 1\,000\,000$ ), broj bakterija koje Mirko treba proizvesti;
- prirodni broj A ( $1 \leq A \leq 1000$ ), broj bakterija koje se stvore pritiskom na tipku „STVORI“.
- prirodni broj B ( $1 \leq B \leq 1000$ ), broj bakterija koje se unište pritiskom na tipku „UNIŠTI“.

**Izlazni podatak:**

- broj P, najmanji mogući ukupni broj pritisaka na tipke stroja tako da bude proizvedeno točno N LGG-bakterija.

**Primjeri:**

RB	Ulaz	Izlaz	Objašnjenje
1.	2 5 4	2	Mirko na početku ima 1 bakteriju. Pritiskom na "STVORI", dobio je još 5 bakterija, pa ih ima ukupno 6. Pritiskom na "UNIŠTI", uništene su 4 bakterije, pa ih ima ukupno 2, koliko ih je i trebao proizvesti. Ukupan broj pritisaka na tipke je 2.
2.	46 30 5	5	Mirko može pritiskati tipke npr. ovim redom: "STVORI", "UNIŠTI", "UNIŠTI", "STVORI", "UNIŠTI".

**Rješenje snimiti pod imenom LGG.BAS ili LGG.PAS ili LGG.C ili LGG.CPP**

U Dubrovniku su odlučili napraviti novo parkiralište. Parkiralište će se sastojati od samo jednog reda parkirnih mjesta. Između svaka dva parkirna mjesta bit će ograda kako bi svaki automobil mogao zauzeti točno jedno mjesto. Svako parkirno mjesto imat će svoj jedinstveni **redni broj**. Parkirno mjesto neposredno do ulaza na parking imat će redni broj 1, mjesto pokraj njega redni broj 2 itd. Posljednje parkirno mjesto imat će redni broj N. Pri ulasku na parkiralište nalazit će se uređaj koji će tiskati kartice na kojima će se, između ostaloga, nalaziti redni broj parkirnog mjesta na koje auto treba parkirati. Uređaj za izdavanje kartica u svakom trenutku zna koja su mjesta zauzeta. Mjesto je zauzeto od trenutka kad je izdana kartica za to mjesto do trenutka dok automobil koji je na tom mjestu stajao nije napustio parkiralište i vratio karticu.

Uređaj će kartice izdavati na sljedeći način:

- **3-park** je slobodno mjesto koje ima slobodno mjesto i s lijeve i s desne strane. Primijetite da prvo i zadnje mjesto nikada nisu 3-park. Ukoliko na parkiralištu postoji 3-park, automobil će parkirati na onaj 3-park koji je najbliži ulazu;
- ako ne postoji 3-park, onda će parkirati na 2-park najbliži ulazu. **2-park** je slobodno mjesto takvo da je i mjesto sa za jedan manjim rednim brojem slobodno;
- ako ne postoji ni 2-park automobil će parkirati na slobodno mjesto najbliže ulazu.



Slika 1: Situacija na parking u nekom trenutku. Zauzeta mjesta su osjenčana

Napiši program koji će pratiti situaciju na parkiralištu (dolazak i odlazak automobila) i u određenom trenutku reći koja su mjesta zauzeta te na koje mjesto treba parkirati automobil koji ulazi na parking.

#### Napomena:

U trenutku kada auto ulazi na parking bit će najmanje jedno slobodno mjesto.

#### Ulazni podaci (tekstualna datoteka *ulaz.txt*):

- u prvom redu tekstualne datoteke *ulaz.txt* nalazi se prirodan broj **N** ( $1 \leq N \leq 1000$ ) – broj parkirnih mjesta na parkiralištu;
- u drugom redu datoteke nalazi se cijeli broj **M** ( $0 \leq M \leq N$ ) – broj zauzetih mjesta na parkiralištu;
- u sljedećih M redova nalaze se prirodni brojevi **X** ( $1 \leq X \leq N$ ) – redni brojevi zauzetih mjesta;
- u sljedećem redu datoteke nalazi se prirodan broj **K** ( $1 \leq K \leq 1000$ ) – broj događaja;
- u sljedećih K redaka nalaze se opisi događaja, pri čemu je svaki događaj string u jednom od sljedećih oblika:
  - "U" – ulazi auto – u tekstualnu datoteku *izlaz.txt* program ispisuje "ULAZ: " i redni broj mjesta na koje auto treba parkirati;
  - "I" **T** – auto koji je stajao na parkirnom mjestu T je napustio parkiralište;
  - "P" – u datoteku se ispisuje "ISPIS: " i **zbroj** rednih brojeva svih zauzetih mjesta na parkiralištu.

#### Izlazni podaci (tekstualna datoteka *izlaz.txt*):

- za svaki nalog ulaza (U) i ispisa (P) program ispisuje odgovarajući rezultat (iz ulaznih podataka).

#### Primjeri:

RB	Ulaz ( <i>ulaz.txt</i> )	Izlaz ( <i>izlaz.txt</i> )	Objašnjenje
1.	13 5 1 3 8 7 9 5 U U I8 P U	ULAZ: 5 ULAZ: 11 ISPIS: 36 ULAZ: 13	<p><i>Primjer odgovara situaciji na slici</i></p> <p><u>1. događaj</u> – automobil ulazi i staje na prvi 3-park, čiji je redni broj <b>5</b>.</p> <p><u>2. događaj</u> – automobil staje na prvi 3-park, na redni broj <b>11</b></p> <p><u>3. događaj</u> – izlazi automobil s broja 8</p> <p><u>4. događaj</u> – ispis – zauzeta mjesta su: 1, 3, 5, 7, 9, 11 a njihov zbroj je <b>36</b>.</p> <p><u>5. događaj</u> – ulazi auto – kako više nema slobodnih 3-park mjesta, auto staje na prvo 2-park mjesto <b>13</b></p>

Tijekom razgledavanja dubrovačkih zidina, Tomislav je uočio tajni prolaz. Prolaskom kroz njega, našao se u pravokutnoj prostoriji. Na zidu je uočio tlocrt, a čim mu se približio, čuo je kako se vrata u prolazu kojim je došao zatvaraju. Uskoro je uočio i slabo vidljiv natpis:

*Putniče!*

*Nalaziš se u Sobi ključeva. Kako bi otključao Tajna vrata i izašao iz Sobe, moraš sakupiti određeni broj ključeva. Svi se ključevi nalaze u ovoj prostoriji i do svih je moguće doći. Na tlocrtu su položaji svih ključeva označeni slovom 'K'. Označena je i tvoja pozicija – potraži na tlocrtu slovo 'X'. Svi zidovi označeni su znakom '#'. Prazna mjesta na karti označena su točkom ('.'). Tajna vrata su jedini izlaz iz prostorije. Sretno!*

Kako Tomislav mora uskoro stići na natjecanje, zanima ga koji je najbrži način da sakupi **točno K** ključeva. Tlocrt prostorije sastoji se od **N** redaka, svaki sa po **M** stupaca. Tomislav se po prostoriji kreće **samo u četiri glavna smjera** – gore, dolje, lijevo i desno. Tlocrt prostorije uvijek će imati zidove na rubovima, tj. neće biti moguće izaći iz prostorije (osim korištenjem Tajnih vrata, koja nisu označena na tlocrtu). Prostorija može imati i pregrade (zidove) u unutrašnjosti. U prostoriji će se nalaziti najviše 8 ključeva.

### Napomena:

U test podacima vrijednim 70 bodova, prostorije **neće** sadržavati unutrašnje zidove (sastojat će se samo od vanjskih zidova).

### Ulazni podaci:

- prirodni broj **N** ( $1 \leq N \leq 30$ ), broj redaka tlocrta;
- prirodni broj **M** ( $1 \leq M \leq 30$ ), broj stupaca tlocrta;
- prirodni broj **K** ( $1 \leq K \leq 8$ ), broj ključeva koje je potrebno sakupiti;
- N redaka, u svakom po M znakova, tlocrt prostorije.

### Izlazni podaci:

- cijeli broj **S**, najmanji broj koraka koje Tomislav mora napraviti kako bi sakupio K ključeva.

### Primjeri:

RB	Ulaz	Izlaz	Objašnjenje
1.	6 10 2 ##### #.X.....K# #.....K...# #.....# #K.....# #####	8	Tomislav može sakupiti 2 ključa ako prvo ode 4 polja udesno, zatim 1 prema dolje, pa 2 udesno, pa 1 prema gore. Ako bi prvo otišao do njemu najbližeg (donjeg lijevog) ključa, tada bi ukupan put bio veći, jer bi do njega napravio 4 koraka, a ostali ključevi su od donjeg lijevog ključa udaljeni 7 i 10 koraka.
2.	6 10 2 ##### #.X#.....K# ##.#.K#...# #..####...# #K.....# #####	14	U nekim primjerima zidovi mogu biti i unutar prostorije.  Sada mu je najbrže pokupiti prvo donji lijevi ključ (4 koraka), a zatim gornji desni (10 koraka), jer je udaljenost do srednjeg ključa sada veća.

**Rješenje snimiti pod imenom ZIDINE.BAS ili ZIDINE.PAS ili ZIDINE.C ili ZIDINE.CPP**