

Lift

Ajapiirang: 10 sekundit

Mälupiirang: 64 MiB (protsessi kohta)

See on suhtluspõhine ülesanne.

On maja, mille korrused on nummerdatud 0 kuni N . Korrus 0 on maapinnal ja korruse N kohal on katus. Seega, katust kaasa arvestades, on majal $N + 2$ tasandit.

Igal korrusel f ($0 \leq f \leq N$) elab täpselt üks elanik ja sellel on salajane täisarv v_f ($0 \leq v_f \leq 3$), mida teab ainult selle korruse elanik. Lisaks elab maja katusel Karlsson. Karlssoni eesmärgiks on määrata kindlaks nii palju salajaste arvude $v_0, v_1, \dots, v_N$ väärtusi kui võimalik. Iga elanik teeb **koostööd**, et Karlsson suudaks välja selgitada võimalikult palju salajaste arvude väärtusi.

Suhtlemiseks kasutavad elanikud hoone lifti järgmisel viisil.

Lift alustab korruselt 0 ja liigub ainult ülespoole. Liftis on nupud, mis on tähistatud korrustega 1 kuni N . Alguses ei ole vajutatud ühtki nuppu. Kui nuppu vajutatakse, jääb see püsivalt vajutatuks. Lift peatub ja avab uksed korrusel f , kui kas $f = 0$ või korrusele f vastavat nuppu on vajutatud mõnel varasemal peatusel. Pärast kõige kõrgema korruse külastamist, mille nuppu on vajutatud (või korruse 0, kui ühtegi nuppu ei vajutata), sõidab lift otse katusele, peatumata ülejäänud korrustel.

Kui lift peatub korrusel f , juhtub järgnev:

1. Selle korruse elanik näeb kõiki nuppe, mis on parajasti alla vajutatud.
2. Ainult selle info ja väärtuse v_f põhjal võib ta vajutada suvalise hulga hetkel vajutamata nuppudest, mis vastavad korrustele, mis on **rangelt kõrgemal** tema enda korrusest. Elanik vajutab neile nuppudele.
3. Lift sõidab üles järgmisele korrusele, millele vastavat nuppu on vajutatud (või katusele, kui kõik vajutatud nuppudele vastavad korrused on külastatud).

Kui lift ei peatu mõnel korrusel, siis selle korruse elanik ei saa ühtki nuppu vajutada.

Kui lift jõuab katusele, näeb Karlsson ainult lõpuks vajutatud nuppude komplekti. Seda teavet kasutades püüab ta taastada võimalikult palju salajasi väärtusi $v_0, v_1, \dots, v_N$.

Salajased väärtused v on fikseeritud enne sinu programmi käivitumist ega muutu protsessi jooksul.

Teostuse üksikasjad

Teste on kokku T .

Esita üks fail, mis realiseerib need kaks funktsiooni.

Esimene funktsioon, mida realiseerida tuleb, on järgmine:

```
std::vector<int> press_buttons(int subtask, int N,  
                               int f, int v, std::vector<int> p)
```

- $subtask$: alamülesanne, milles seda funktsiooni välja kutsutakse, kus $0 \leq subtask \leq 4$;
- N : viimase korruse number;
- f : praeguse korruse number, kus $0 \leq f \leq N$;
- v : korrusele number f kirjutatud salajane arv v_f ;
- p : hetkel vajutatud nupud kasvavas järjekorras.

Seda funktsiooni kutsutakse välja, kui lift avab ukseid korrusel f (kui kas $f = 0$ või korrusele f vastavat nuppu on vajutatud mõnel varasemal peatusel). Seda funktsiooni ei kutsuta välja konkreetse korruse jaoks, kui vastavat nuppu ei ole varem vajutatud.

Tagastatav väärtus on loend uutest nuppudest, mida vajutada. Tagastatavas loendis olevate väärtuste järjekord ei ole oluline. Iga vajutatav nupp x peab rahuldama järgmisi tingimusi:

- $f < x \leq N$;
- x ei ole juba loendis p ning x esineb tagastatavas loendis täpselt üks kord.

Teine funktsioon, mida realiseerida tuleb, on järgmine:

```
std::vector<int> answer(int subtask, int N, std::vector<int> p)
```

- $subtask$: alamülesanne, milles seda funktsiooni välja kutsutakse, kus $0 \leq subtask \leq 4$;
- N : viimase korruse number;
- p : lõpuks vajutatud nuppude loend kasvavas järjekorras.

Seda funktsiooni kutsutakse igal testil välja täpselt üks kord – siis, kui lift jõuab katusele.

Tagastatav loend peab olema pikkusega $N + 1$, kus element kohal i peab olema võrdne väärtusega v_i , samas väärtused, mida Karlsson **ei suuda taastada**, tuleks asendada väärtusega -1 .

Tähtis: Inimesed hoones ei saa omavahel suhelda muul viisil kui vajutatud nuppude kaudu. Selleks käivitatakse sinu programm $N + 2$ eraldi protsessina – üks iga korruse jaoks ja üks katuse jaoks. Igas korruste protsessis on maksimaalselt T funktsiooni `press_buttons` väljakutset ja katuse

protsessis on täpselt T funktsiooni `answer` väljakutset. Seega ei tohiks sinu programm eeldada ühist olekut (näiteks globaalseid muutujaid). Igal protsessil on 64 MiB mälu, mis on eraldatud teiste protsesside mälust, ja kõik need väljakutsed (kuni $(N + 1) \times T_{\text{press_buttons}}$ kutset ja täpselt T_{answer} kutset) peavad lõppema 10 sekundi jooksul.

Piirangud

- $N = 60$
- $T \leq 10\,000$
- $0 \leq v_i \leq 3$ iga $0 \leq i \leq N$ korral

Näited

Näidistestis on 1 test järgnevate korruste salajaste väärtustega:

```
1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 1
1 0 1 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1
```

Siin on näide suhtlusest:

Võistleja programm	Žürii programm
	<code>press_buttons(0, 60, 0, 1, {})</code>
<code>return {2}</code>	
	<code>press_buttons(0, 60, 2, 0, {2})</code>
<code>return {13, 42}</code>	
	<code>press_buttons(0, 60, 13, 0, {2, 13, 42})</code>
<code>return {}</code>	
	<code>press_buttons(0, 60, 42, 1, {2, 13, 42})</code>
<code>return {}</code>	
	<code>answer(0, 60, {2, 13, 42})</code>
<code>return {1, -1, 0, -1, -1, ..., -1}</code>	

See suhtlus leiab korruste 0 ja 2 salajased väärtused. Kõik muud tagastatud väärtused on võrdsed -1 -ga, sest Karlsson ei suutnud neid leida.

Alamülesanded

Alamülesanne	Punktid	Täiendavad piirangud	Täispunktideks vajalik leitud salajaste väärtuste arv
0	0	Näide.	–
1	15	$0 \leq v_i \leq 1$ $v_0 = v_N = 1$ kui $v_i = 0$, siis $v_{i+1} = 1$ iga $0 \leq i < N$ korral	60
2	35	$0 \leq v_i \leq 1$	40
3	15	$0 \leq v_i \leq 2$	30
4	35	$0 \leq v_i \leq 3$	25

Hindamine

Kui sinu programm tagastab korruse jaoks vale salajase väärtuse või teostab lubamatu toimingut (näiteks tagastab vale pikkusega `vector` objekti) mõnes alamülesande testis, saab see lahendus selle alamülesande eest null punkti.

Ühe testi jaoks on sinu lahenduse *leitud salajaste väärtuste arv* nende positsioonide arv, mille tagastatud väärtus ei ole -1 . Olgu K alamülesande kõigis testides leitud salajaste väärtuste arvude miinimum (igal alamülesandel on ainult 1 test, mis aga sisaldab kuni 10 000 testi). Seejärel arvutatakse iga alamülesande punktid järgmise tabeli alusel.

Alamülesanne	K vahemik	Punktid
0	-	0
1	$K < 60$	$0.25K$
	$K \geq 60$	15
2	$K < 30$	$0.5K$
	$30 \leq K < 40$	$2K - 45$
	$K \geq 40$	35
3	$K < 30$	$0.5K$
	$K \geq 30$	15
4	$K < 20$	$0.7K$
	$20 \leq K < 25$	$4K - 66$
	$K \geq 25$	35

Näidishindaja

Näidishindaja teeb kõik funktsioonikutsed kõigi testide jaoks ühe käivituse jooksul.

Sisendi vorming on järgmine:

- rida 1: kolm täisarvu – testide arv T , viimase korruse number N ja alamülesande number S ;
- rida $1 + i$: $N + 1$ täisarvud $v_0, v_1, \dots, v_N$, kus v_f on korrusele f kirjutatud salanumbri väärtus selle testi jaoks.

Väljundi vorming on järgmine:

- rida i : üks täisarv – korrektselt leitud salajaste väärtuste v arv testis i ;
- rida $T + 1$: lõplikud punktid.

Üksikasjalikuma tagasiside saamiseks muuda makro `DETAILED` hindamisprogrammi esimeses reas väärtusest `false` väärtuseks `true`.

Sa saad lasta hindajal enda jaoks teste (salanumbrite väärtusi v_f) genereerida, muutes hindaja teisel real makro `AUTO_GENERATE` väärtusest `false` väärtuseks `true`.