

Elevator

Временско ограничување: 10 секунди

Мемориско ограничување: 64 MiB (по процес)

Ова е задача за комуникација.

Постои зграда со катови нумерирани од 0 до N . Кат 0 е приземјето, а над катот N се наоѓа подкровието. Значи, вклучувајќи го и подкровието, зградата има $N + 2$ нивоа.

Секој кат f ($0 \leq f \leq N$) е населен од точно еден жител и има таен цел број v_f ($0 \leq v_f \leq 3$), познат само на неговиот жител. Освен тоа, постои Карлсон кој живее на подкровието на оваа зграда. Целта е Карлсон да утврди колку што може повеќе од вредностите $v_0, v_1, \dots, v_N$, додека секој жител **соработува** за да го максимизира бројот на вредности што тој може да ги открие.

За да комуницираат, жителите го користат лифтот на зградата на следниов начин.

Лифтот започнува на кат 0 и патува само нагоре. Внатре во лифтот има копчиња означени со катовите од 1 до N . Првично, ниту едно копче не е притиснато. Штом едно копче е притиснато, тоа останува притиснато трајно. Лифтот застанува и се отвора на кат f ако $f = 0$, или ако копчето што одговара на катот f било притиснато на некое претходно застанување. По посетата на најгорниот кат чие копче е притиснато (или кат 0 ако никогаш не е притиснато ниту едно копче), лифтот продолжува директно до подкровието без да застанува на преостанатите катови.

Кога лифтот застанува на кат f , се случува следново:

1. Жителот го гледа целосниот сет копчиња кои се моментално притиснати.
2. Врз основа само на оваа информација и вредноста v_f , тој може да избере кое било подмножество од моментално непритиснатите копчиња кои одговараат на катови **строго над** нивниот кат. Жителот ги притиска овие копчиња.
3. Лифтот оди нагоре до следниот најнизок кат за кој соодветното копче е притиснато (или до подкровието ако сите катови чии копчиња се притиснати веќе се посетени).

Ако лифтот не застане на некој кат, соодветниот жител не може да притисне никакви копчиња.

Кога лифтот ќе стигне до подкровието, Карлсон го гледа само конечниот сет притиснати копчиња. Користејќи ја оваа информација, тој се обидува да открие колку што е можно

повеќе од вредностите $v_0, v_1, \dots, v_N$.

Вредностите на v се фиксирани пред да започне вашата програма и не се менуваат за време на процесот.

Детали за имплементацијата

Ќе има T тест случаи.

Испратете една датотека што ги имплементира следниве две функции.

Првата функција што треба да ја имплементирате е следнава:

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

- $subtask$: подзадачата за која се повикува оваа функција, каде $0 \leq subtask \leq 4$;
- N : бројот на последниот кат;
- f : тековниот кат, каде $0 \leq f \leq N$;
- v : вредноста v_f запишана на катот f ;
- p : тековно притиснатите копчиња во растечки редослед.

Оваа функција се повикува кога лифтот се отвора на катот f (ако $f = 0$, или ако копчето што одговара на катот f било притиснато на некое претходно застанување). Оваа функција нема да биде повикана за конкретен кат ако соодветното копче не било притиснато претходно.

Повратната вредност е листата на нови копчиња што треба да се притиснат. Редоследот на вредностите во вратената низа не е важен. Секое притиснато копче x мора да го задоволува следново:

- $f < x \leq N$;
- x не се наоѓа веќе во p и x се појавува точно еднаш во вратената низа.

Втората функција што треба да ја имплементирате е следнава:

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

- $subtask$: подзадачата за која се повикува оваа функција, каде $0 \leq subtask \leq 4$;
- N : бројот на последниот кат;
- p : конечниот сет притиснати копчиња во растечки редослед.

Оваа функција се повикува точно еднаш по тест случај кога лифтот ќе стигне до подкровието.

Вратената низа мора да има должина $N + 1$, i -тиот елемент од неа треба да биде еднаков на v_i , додека вредностите кои Карлсон **не може да ги открие** треба да бидат заменети со -1 .

Важно: Луѓето во зградата не можат да комуницираат на никаков друг начин освен преку притиснатите копчиња. За да се обезбеди ова, вашата програма ќе биде извршена како $N + 2$ одделни процеси, по еден за секој кат, и еден за подкровието. Во секој процес за катовите, ќе има најмногу T повици на функцијата `press_buttons`, а во процесот за подкровието – точно T повици на функцијата `answer`. Затоа вашата програма не смее да претпоставува никаква споделена состојба (на пример глобални променливи). Секој процес ќе има на располагање 64 MiB меморија, одделна од меморијата на другите процеси, и сите тие повици (до $(N + 1) \times T$ повици на `press_buttons` и точно T повици на `answer`) треба да завршат за 10 секунди.

Ограничувања

- $N = 60$
- $T \leq 10\,000$
- $0 \leq v_i \leq 3$ за секое $0 \leq i \leq N$

Пример

Тест-примерот има 1 тест случај со следниве вредности на катовите:

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

Еве пример-интеракција:

Вашата програма	Судиската програма
	<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>	

Оваа интеракција коректно ги открива 0-тата и 2-рата вредност. Сите останати вратени вредности се еднакви на -1 бидејќи Карлсон не ги открил.

Подзадачи

Подзадача	Поени	Дополнителни ограничувања	Број на катови за откривање за целосни поени
0	0	Примерот.	–
1	15	$0 \leq v_i \leq 1$ $v_0 = v_N = 1$ Ако $v_i = 0$, тогаш $v_{i+1} = 1$ за секое $0 \leq i < N$	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

Бодување

Ако вашата програма врати неточно откриена вредност или изврши невалидна операција (на пример врати `vector` со погрешна должина) во кој било од тест случаите за некоја подзадача, вашето решение ќе добие нула поени за таа подзадача.

За еден тест случај, *бројот на откриени вредности* на вашето решение е бројот на позиции чија вратена вредност не е -1 . Нека K е минималниот број на откриени вредности меѓу сите тест случаи на една подзадача (секоја подзадача има само 1 тест кој содржи до 10 000 тест случаи). Тогаш, поените за секоја подзадача се пресметуваат според табелата подолу.

Подзадача	Опсег на K	Поени
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