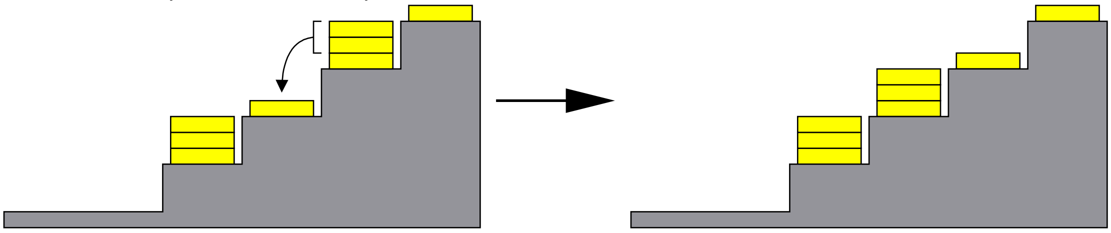
Submit:	`stairs.c` / `stairs.cpp` / `stairs.pas`	Časový limit:	on the order of seconds
Vstup:	(none)	Paměťový limit:	32 MB
Výstup:	(none)	Bodů:	100

Když jsou studenti FMFI UK v Bratislavě nuceni čekat na schodech, obvykle si krátí čas následující hrou: Hra je určena pro dva hráče. Na počátku hry hráči položí na schody všechny mince, které mají. Pak střídavě táhnou. V každém tahu si hráč vybere jeden schod a přesune libovolný počet mincí z vybraného schodu o jeden schod níže. Hráč, který přesune poslední minci na podlahu pod schody, získává všechny mince.



Příklad jednoho tahu – student přesunul dvě mince ze třetího schodu

Úloha: Napište program, který bude hrát tuto hru. Bude vám poskytnuta knihovna (`uStairs.pas` nebo `uStairs.h`) hrající jako váš oponent. S knihovnou se komunikuje prostřednictvím následujících funkcí:

Pascal:

Přidejte řádek `uses uStairs;` na začátek programu.

```
procedure Init_game(var n:longint; var stairs:TARRAY)      TARRAY=array[0..1000] of longint
procedure Turn(from,coins:longint; var hisfrom,hiscoins:longint)
```

C:

Přidejte řádek `#include "uStairs.h"` na začátek programu.

```
void Init_game(int *n, int *stairs)
void Turn(int from, int coins, int *hisfrom, int *hiscoins)
```

Doporučujeme Vám kompilovat programy pomocí skriptu `compile`, který se postará o správné slinkování s knihovnou.

Funkci `Init_game` musíte volat právě jednou – na počátku hry. Tato funkce vám vrátí počet schodů ve hře (v proměnné n) a počet mincí ležících na každém schodu (v poli *stairs*). Počet schodů je mezi 1 a 1 000 včetně. Schody jsou číslovány od nuly (podlaha pod schody) po n. Počáteční počet mincí na každém schodu bude mezi 0 a 1 000 včetně.

Tahy provádíte voláním funkce `Turn`. V parametrech *from* a *coins* jí dáte číslo vybraného schodu a počet mincí, které chcete přesunout na nižší schod (tedy schod s číslem $from - 1$). Knihovna vám vrátí svůj tah ve „výstupních“ parametrech *hisfrom* a *hiscoins*.

Váš program nesmí skončit. To za vás udělá knihovna v okamžiku, kdy zjistí, že jste vyhráli nebo prohráli.

Vstup a výstup: Váš program by neměl nic číst ani vypisovat. Hru hraje pouze komunikací s knihovnou. Můžete předpokládat, že v průběhu vyhodnocování bude knihovna hrát optimálně. Váš program by měl pozice, které mohou být vyhrané, vyhrát, a v ostatních hrát korektně a prohrát.

Příklad hry: $n = 4$, *stairs*$= (47, 2, 1, 0, 1)$.

Hráč	Tah	Situace po tahu				
1.	1 mince ze 4. schodu	47	2	1	1	0
2.	1 mince z 2. schodu	47	3	0	1	0
1.	1 mince ze 3. schodu	47	3	1	0	0
2.	2 mince z 1. schodu	49	1	1	0	0
1.	1 mince z 1. schodu	50	0	1	0	0
2.	1 mince ze 2. schodu	50	1	0	0	0
1.	1 mince z 1. schodu	51	0	0	0	0

Hráč 1 vyhrál.