Zadatak: DeCSS

Content Scramble System (CSS) je metoda kojom se kriptira sadržaj na DVD medijima kako bi mu mogli pristupati samo licencirani uređaji. U varijanti ovog sustava kojom se bavimo, *ključ* K je niz od točno 42 bita $k_1, k_2, \ldots, k_{42}$, dok je *sadržaj* niz od n byte-ova. Sadržaj se *kriptira* tako da se najprije na temelju ključa generira takozvani *tok ključa* $T(K)$ – također niz od n byte-ova te se, zatim, primijeni operacija bitovni ekskluzivni ILI na odgovarajuće elemente sadržaja i toka ključa.

Ako je poznat kriptirani tekst, te neki byte-ovi sadržaja, onda možemo odrediti odgovarajuće byte-ove toka ključa. Vaš je zadatak da na temelju *djelomično poznatog* toka ključa $T(K)$ odredite jedan mogući ključ K.

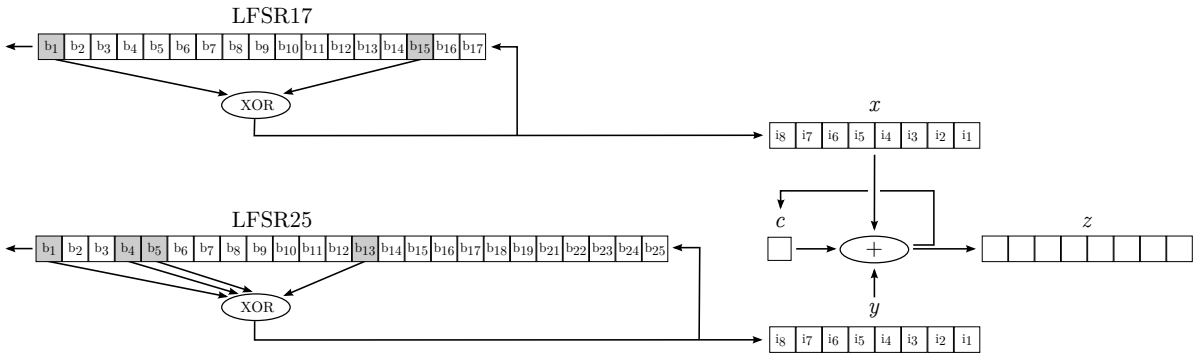
Funkcija $T(K)$ koja generira tok ključa se zasniva logičkom sklopu *Linear-feedback shift register* (LFSR). Stanje LFSR-a se sastoji od m bitova $b_1, b_2, \ldots, b_m$, a funkcionalnost se definira tako da se odredi skup *povratnih* pozicija. U jednom *ciklusu* LFSR generira jedan bit izlaza te promjeni stanje na sljedeći način:

1. Izračuna se *izlazni bit* b tako da se zbroje bitovi na povratnim pozicijama. Ako je rezultat paran onda je b jednak 0, a inače je jednak 1. Dakle, b je ekskluzivno ILI bitova na povratnim pozicijma.
2. Svi bitovi stanja se pomiču ulijevo, bit b_1 se odbacuje, dok se bit b postavlja na zadnju poziciju. Dakle novo stanje je $b_2, b_3, \ldots, b_m, b$.

Jedan *korak* LFSR-a se sastoji od 8 ciklusa, a rezultat je jedan byte koji čine izlazni bitovi ciklusa redom s desna nalijevo. Točnije, ako su izlazni bitovi ciklusa redom $i_1, i_2, \ldots, i_8$, onda je rezultat koraka cijeli broj između 0 i 255 čiji je binarni zapis $(i_8 i_7 \ldots i_1)_2$.

CSS koristi dva ovakva sklopa:

- LFSR17 veličine 17 bitova u kojem su pozicije 1 i 15 označene kao povratne, a početno stanje čine redom bitovi ključa $k_1, k_2, \ldots, k_{17}$.
- LFSR25 veličine 25 bitova u kojemu su pozicije 1, 4, 5, i 13 označene kao povratne, a početno stanje čine redom bitovi ključa $k_{18}, k_{19}, \ldots, k_{42}$.



Slika 1: Generiranje toka ključa

Tok ključa $T(K)$ se generira na sljedeći način:

1. Sklopovi LFSR17 i LFSR25 se postave u početno stanje pomoću ključa K na opisani način.
2. Vrijednost varijable c se postavi na 0.
3. Ponavlja se n puta:
   (a) Izvede se jedan korak sklopa LFSR17, neka je x rezultat koraka.
   (b) Izvede se jedan korak sklopa LFSR25, neka je y rezultat koraka.
   (c) Izračuna se zbroj $z = x + y + c$
   (d) Ukoliko je $z \geq 256$, z se umanji za 256, a c se postavi na 1. Inače se c postavi na 0.
   (e) Sljedeći byte toka ključa je upravo z.

Zadan je tok ključa u kojem su neki byte-ovi poznati, a neki su nepoznati. Odredite jedan mogući ključ K od kojeg se na opisani način može dobiti tok koji odgovara zadanom.

Ulazni podaci

Prvi red sadrži prirodni broj n, duljinu zadanog toka ključa. Sljedeći red sadrži n cijelih brojeva $t_1, t_2, \ldots, t_n$ – redom byte-ovi toka ključa. Ukoliko je k-ti byte nepoznat vrijedi $t_k = -1$, a ukoliko je poznat vrijedi $0 \leq t_k \leq 255$.

Izlazni podaci

U prvi red ispišite bitove traženog ključa $k_1, k_2, \ldots, k_{42}$ bez razmaka.

Napomena: Rješenje će uvijek postojati, iako ne mora biti jedinstveno.

Bodovanje

Ovo je *zadatak samo s izlazom* – ne šaljete izvorni kod na evaluaciju već je samo potrebno riješiti 10 test podataka koje možete preuzeti sa sustava za evaluaciju. Opisi test podataka su sadržani u sljedećoj tablici.

Podzadatak	Ulazna datoteka	Duljina toka ključa	Broj poznatih byte-ova	Broj bodova
1	`decss.in.1`	20	20	8
2	`decss.in.2`	25	21	8
3	`decss.in.3`	20	8	8
4	`decss.in.4`	30	10	8
5	`decss.in.5`	31	8	9
6	`decss.in.6`	30	20	9
7	`decss.in.7`	200	9	12
8	`decss.in.8`	300	12	12
9	`decss.in.9`	500	12	13
10	`decss.in.10`	500	12	13

Primjeri test podataka

ulaz

```
7
154 39 225 99 151 145 -1
```

izlaz

```
01100011001010010110011000000000000000111011
```

Pojašnjenje prvog primjera: Sljedeća tablica sadrži detalje generiranja prva četiri byte-a toka ključa. Prvi redak tablice sadrži početno stanje, a svi ostali redci stanje neposredno nakon završetka određenog ciklusa odnosno koraka. U svakom LFSR-u su sivom bojom označene povratne pozicije te je podcrtan posljednji bit koji je ujedno bio i izlazni bit u tom ciklusu.

Korak	Ciklus	LFSR17	LFSR25	x	y	c	z
Početno stanje		0 1100 0110 0101 0010	1 1001 1000 0000 0000 0011 1011			0	
	1	1 1000 1100 1010 0100	1 0011 0000 0000 0000 0111 0110				
	2	1 0001 1001 0100 1000	0 0110 0000 0000 0000 1110 1101				
	3	0 0011 0010 1001 0001	0 1100 0000 0000 0001 1101 1011				
	4	0 0110 0101 0010 0010	1 1000 0000 0000 0011 1011 0110				
	5	0 1100 1010 0100 0100	1 0000 0000 0000 0111 0110 1101				
	6	1 1001 0100 1000 1001	0 0000 0000 0000 1110 1101 1011				
	7	1 0010 1001 0001 0011	0 0000 0000 0001 1101 1011 0110				
	8	0 0101 0010 0010 0111	0 0000 0000 0011 1011 0110 1101				
1				228	182	1	**154**
	9	0 1010 0100 0100 1111	0 0000 0000 0111 0110 1101 1011				
	10	1 0100 1000 1001 1111	0 0000 0000 1110 1101 1011 0111				
	11	0 1001 0001 0011 1110	0 0000 0001 1101 1011 0110 1110				
	12	1 0010 0010 0111 1101	0 0000 0011 1011 0110 1101 1101				
	13	0 0100 0100 1111 1010	0 0000 0111 0110 1101 1011 1011				
	14	0 1000 1001 1111 0100	0 0000 1110 1101 1011 0111 0110				
	15	1 0001 0011 1110 1001	0 0001 1101 1011 0110 1110 1101				
	16	0 0010 0111 1101 0011	0 0011 1011 0110 1101 1101 1010				
2				203	91	1	**39**
	17	0 0100 1111 1010 0110	0 0111 0110 1101 1011 1011 0100				
	18	0 1001 1111 0100 1101	0 1110 1101 1011 0111 0110 1001				
	19	1 0011 1110 1001 1011	1 1101 1011 0110 1110 1101 0010				
	20	0 0111 1101 0011 0111	1 1011 0110 1101 1101 1010 0100				
	21	0 1111 1010 0110 1111	1 0110 1101 1011 1011 0100 1000				
	22	1 1111 0100 1101 1111	0 1101 1011 0111 0110 1001 0001				
	23	1 1110 1001 1011 1110	1 1011 0110 1110 1101 0010 0010				
	24	1 1101 0011 0111 1100	1 0110 1101 1101 1010 0100 0101				
3				62	162	0	**225**
	25	1 1010 0110 1111 1000	0 1101 1011 1011 0100 1000 1011				
	26	1 0100 1101 1111 0001	1 1011 0111 0110 1001 0001 0110				
	27	0 1001 1011 1110 0011	1 0110 1110 1101 0010 0010 1101				
	28	1 0011 0111 1100 0110	0 1101 1101 1010 0100 0101 1011				
	29	0 0110 1111 1000 1100	1 1011 1011 0100 1000 1011 0111				
	30	0 1101 1111 0001 1001	1 0111 0110 1001 0001 0110 1111				
	31	1 1011 1110 0011 0010	0 1110 1101 0010 0010 1101 1110				
	32	1 0111 1100 0110 0101	1 1101 1010 0100 0101 1011 1101				
4				166	189	1	**99**