

# Codewar

## Obiective

În urma realizării acestei teme, studentul va fi capabil:

- să utilizeze matrici și vectori
- să folosească operații pe biți
- să înțeleagă conceptul de criptare/ decriptare

## Enunțul temei

Locuitorilor din Codeland le-a fost declarat război de către nemiloșii barbari din Common Town, mult mai numeroși și specializați în lupte. Întrucât printre codelandieni există un vrăjitor bătrân, fost student al Facultății de Automatică și Calculatoare, acesta le-a promis că își va folosi ultimele puteri pentru a vă trimite vouă, studenților, locațiile bazelor în care sunt situați luptătorii din Codeland, pentru ca voi să vă alăturați lor în cele mai sigure baze. Magia, însă, e mai complicată de atât, iar voi, studenții, trebuie să vă arătați demni de a o accesa! Datele transmise de vrăjitor ajung la voi criptate, alături de un pergament pe care sunt explicate regulile decriptării.

### Task 1

*40% din punctajul pe teste*

Primul număr primit,  $b$ , reprezintă ingredientul principal al magiei. Acesta este urmat de dimensiunile terenului pe care sunt situate bazele,  $m$  și  $n$ , ce este structurat asemenea unei matrice. Terenul conține linii între  $0$  și  $m-1$  și coloane între  $0$  și  $n-1$ . Până la întâlnirea lui  $-1$ , vă va fi oferit un șir de numere, pe care voi trebuie să îl decodificați;  $b$  reprezintă numărul maxim de biți pe care îi puteți reține deodată.

Considerând că  $b$  este egal cu 3, iar șirul de numere este 2, 3, 1, 5, -1 (în binar, 10, 11, 1, 101), șirul nou format va fi 1 0 1 1 1 0, astfel:

- luăm cei 2 biți ai numărului 2, deoarece nu ajung la limita lui  $b$ ; (1 0)
- din cei 2 biți ai numărului 3, îl putem lua doar pe primul, moment în care am atins numărul maxim de biți pe care îi putem reține deodată; (1 0 1)
- cel de-al doilea bit al numărului 3 este ignorat;
- luăm singurul bit al numărului 1; (1 0 1 1)
- luăm doar primii 2 biți ai numărului 5 și îl ignorăm pe ultimul, deoarece ar depăși limita impusă de  $b$ . (1 0 1 1 1 0)

Prima parte a acestei teme presupune găsirea acestui șir de biți, folosindu-vă de numărul  $b$  și de șirul de numere oferite de vrăjitor.

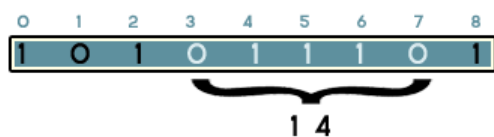
### Task 2

*30% din punctajul pe teste*

Din noul șir pot fi extrase coordonatele bazelor în care sunt situați soldații; până la întâlnirea perechii  $\{-1, -1\}$ , vă sunt oferite intervale de tipul  $[x, y]$ , ce trebuie interpretate astfel:

- numărul format din biții situați în intervalul  $[x, y]$  reprezintă poziția în matrice;

Considerăm șirul de biți 1 0 1 0 1 1 1 0 1,  $x = 3$ ,  $y = 7$ ,  $m = 4$ ,  $n = 5$ . Numărul format va fi 14 (0 1 1 1 0), iar corespondentul lui în matrice este ilustrat în imaginea de mai jos.



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

- în cazul în care  $y < x$ ,  $x \geq \text{dimensiune\_sir\_biti}$  sau  $y < 0$  atunci perechea  $\{x, y\}$  se ignora și se trece la urmatoarea;
- în cazul în care  $x < 0$  atunci  $x = 0$ ;
- în cazul în care  $y \geq \text{dimensiune\_sir\_biti}$  atunci  $y = \text{poziția ultimului bit din șir} (\text{dimensiune\_sir\_biti} - 1)$ ;
- în cazul în care  $y - x > 30$ ,  $y$  se va reduce cât este necesar pentru a forma un număr de maxim 31 biți ( $\leftarrow 2147483647$ );
- în cazul în care din intervalul  $[x, y]$  se formează un număr care depășește dimensiunile matricei, numărul va fi redus la valoarea lui modulo  $m \cdot n$ .

Pentru a obține punctajul celui de-al doilea task, trebuie să afișați pe un rând, separate prin spațiu, coordonatele bazelor.

## Task 3

*30% din punctajul pe teste*

Dupa ce ați găsit coordonatele bazelor în care sunt situați soldații, trebuie să găsiți cea mai sigură "reuniune de baze". O reuniune de baze reprezintă mai multe baze cu proprietatea că din oricare se poate ajunge la oricare trecând numai prin alte baze. Trecerea dintr-o bază în alta se poate face în oricare din direcțiile sus, jos, stânga sau dreapta (nu și pe diagonală). Siguranța este reprezentată de numărul de baze incluse într-o reuniune. Considerăm că, în matrice, o bază se identifică prin cifra 1, iar absența bazei în acea poziție este dată de cifra 0.

## Date de intrare

De la tastatură veți primi pe prima linie trei variabile  $b$ ,  $m$  și  $n$ , reprezentând cheia de decryptare, numărul de linii ale terenului de război și numărul de coloane. Pe următoarea linie veți primi un șir de numere care se va încheia cu -1. Pe următoarele linii veți primi

câte două numere, x și y, cu semnificația de mai sus, iar pe ultima linie veți găsi perechea [-1, -1].

## Date de ieșire

În fișierul "codewar.out" veți afișa pe prima linie șirul de biți găsit în urma îndeplinirii TASK-ului 1, pe linia următoare toate numerele formate la TASK 2, iar pe ultima linie coordonatele celulei cea mai din sus-stanga din cea mai sigură reuniune și numărul de baze din care este alcătuită. Dacă sunt mai multe reuniuni cu același număr maxim de baze, se vor afișa toate, pe linii diferite. Formatul este următorul: SIR\_DE\_BITI SIR\_DE\_NUMERE LIN COL NR\_BAZE LIN COL NR\_BAZE ...

## Exemplu

### Input

```
5 4 4
2 19 8 1 6 -1
1 2
1 4
2 5
8 17
-1 4
5 19
2 3
-1 -1
```

### Output

```
1 0 1 0 0 1 0 0 0 1 1 1 0
1 4 9 14 4 14 2
0 1 2
```

### Explicații

În binar, șirul devine 10, 10011, 1000, 1, 110. Șirul nou format va fi 1 0 1 0 0 1 0 0 0 1 1 1 0, astfel:

- luăm cei 2 biți ai numărului 2, deoarece nu ajung la limita lui b; (1 0)

- din cei 5 biți ai numărului 19, îi putem lua doar pe primii 3, moment în care am atins numărul maxim de biți pe care îi putem reține deodată; (1 0 1 0 0)
- ultimii doi biți ai numărului 19 sunt ignorați;
- luăm cei 4 biți ai numărului 8, deoarece nu ajung la limita lui b; (1 0 1 0 0 1 0 0 0)
- luăm singurul bit al numărului 1 și completăm un singur bit rămas până la limita de b biți; (1 0 1 0 0 1 0 0 0 1)
- luăm toți biții numărului 6, deoarece nu ajung la limita lui b; (1 0 1 0 0 1 0 0 0 1 1 1 0)

Pozițiile în matrice sunt:

- 1 (0 1)
- 4 (0 1 0 0)
- 9 (1 0 0 1)
- 14 (0 1 1 1 0)
- $20 \% (4*4) = 4$  (1 0 1 0 0)
- $128 + 8 + 4 + 2 = 142 \% (4*4) = 14$  (1 0 0 0 1 1 1 0)
- 2 (1 0)

Matricea va arăta astfel:

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

Cea mai sigură bază este dată de reuniunea celor două așezări de pe prima linie (linia 0), începând cu a doua coloana (coloana 1).

## Precizări legate de implementarea temei

- $1 < m, n < 400$
- $0 < b < 31$
- Tema se va implementa în limbajul C;
- Fișierul de implementare a temei se va numi `codewar.c`;
- Fișierele temei trebuie OBLIGATORIU împachetate într-o arhivă de tip '.zip', cu numele 'Grupa\_NumePrenume\_Tema2.zip'. Această arhivă va conține:
  - Codul sursă al programului vostru.
  - Un fișier `Makefile` care să conțină regulile `build` și `clean`. Regula `build` va compila programul într-un executabil cu numele `codewar`. Regula `clean` va

șterge executabilul și eventual toate binarele intermediare (fișiere obiect) generate de voi.

- Un fișier `README` care să conțină explicații privitoare la modul de rezolvare;
- Arhiva temei NU va conține fișiere binare;
- Arhiva va fi trimisă atât pe [vmchecker](#) cât și pe [moodle](#).
- O temă care nu compilează nu va fi punctată.
- Deși programul vostru va trebui să citească direct de la tastatură și să afișeze pe ecran (folosind, de exemplu, `scanf` și `printf`), puteți să citiți datele și să le scrieți în fișiere, folosind redirectările din consolă, fără să modificați programul. Dacă fișierul de intrare este `codewar.in` și cel de ieșire este `codewar.out`, iar programul vostru se numește `codewar`, veți scrie în terminal:

```
./codewar < codewar.in > codewar.out
```