

3. (2p) Scrieți programul C complet!

a) Implementați o funcție *nr_cuvinte_distincte* care primește ca și parametru o frază în care cuvintele apar separate prin spațiu, virgulă, punct, semnul întrebării, semnul exclamării și care returnează numărul de cuvinte distincte pe care îl are fraza respectivă (se consideră cuvânt duplicat chiar dacă el apare scris diferit din cauza folosirii unor majuscule/minuscule – exemplu: *Cine/cine*).

b) Implementați apoi funcția *main* în care se citește un număr natural pozitiv *n* și apoi *n* fraze și care utilizând funcția *nr_cuvinte_distincte* determină și afișează la ieșirea standard fraza care are numărul maxim de cuvinte distincte. În cazul în care există mai multe astfel de fraze, se va afișa ultima.

Constrângeri: lungimile șirurilor de caractere nu depășesc 200; numărul de fraze $1 \leq n \leq 100$

Exemplu:

Citire de la intrarea standard:	Afișare la ieșirea standard:
4 Haideti la examen, la marele examen! Cine si cu cine vine azi la cine? Zi geroasa de inceput de an nou. Examen greu, examen usor, studentii stiu!	Fraza numarul 3 are 6 cuvinte distincte.

4. (1p) Șirul Thue-Morse este definit în mod recursiv: $T_0 = 0$ și pentru *n* pozitiv T_n este format din concatenarea biților lui T_{n-1} și a biților negați lui T_{n-1} . Scrieți funcția recursivă care returnează al *n*-lea termen sub forma unui număr întreg pozitiv. Constrângeri: $0 \leq n \leq 6$. Exemplu: $T_2 = 6 = 0110_{(2)}$.

5. (3p) În jocul *Connect-k* se plasează jetoane colorate pe o tablă cu găuri așezată în picoare. Dacă se formează un șir de *k* jetoane de aceeași culoare, ele se pot extrage de pe tablă.

a) Citiți datele de intrare din fișierul text *input.txt* care descrie starea inițială a tablei (dimensiuni și elemente). Găsiți pe primul rând valorile *n* și *k*. Numărul *n* reprezintă dimensiunea tablei care poate fi privită ca o matrice pătratică (*n* x *n*), iar *k* reprezintă parametrul specificat în definiție. Apoi, pe următoarele *n* rânduri găsiți matricea *a* care reprezintă tabla. Valorile $a[i][j]$ nenule 1, 2 sau 3 reprezintă jetoane de o culoare specifică, iar 0 reprezintă o casuță goală. Memorați datele de intrare și scrieți apoi o funcție de afișare a lor la ieșirea standard.

Constrângeri: $0 < n, k \leq 1000$, $0 \leq a[i][j] < 4$

b) Implementați funcția care primește starea tablei, un parametru care indică direcția și parametrul *k*. Direcția poate fi orizontală, verticală, diagonală 1, diagonală 2. Se va verifica dacă există *k* jetoane consecutive în direcția dată undeva pe tablă. Dacă da, atunci se șterg toate aceste jetoane din acea zonă (în cazul în care există mai multe zone, se va considera doar una). După ștergere restul jetoanelor cad în jos. Funcția returnează starea tablei după modificare. Afișați, la ieșirea standard, în funcția *main* conținutul tablei după aplicarea operațiilor menționate.

Exemplu: fișier de intrare și rezultate după două operații diferite (nu sunt succesive).

input.txt	după apel cu diagonală 1, k = 3	după apel cu vertical, k = 3
5 3 0 0 1 0 0 2 1 3 1 0 1 2 3 2 1 1 2 3 2 1 3 1 3 1 2	0 0 0 0 0 0 0 0 0 0 2 1 3 0 0 1 2 3 2 0 1 2 3 2 1 3 1 3 1 2	0 0 0 0 0 2 1 0 1 0 1 2 0 2 1 1 2 1 2 1 3 1 3 1 2

c) Utilizând tipuri de date fundamentale din C pentru stocarea stării tablei, salvați aceasta într-un fișier binar numit *ouput.bin* astfel încât acesta să ocupe cât mai puțin spațiu pe dispozitivul de stocare. Deschideți apoi fișierul binar, determinați și afișați la ieșirea standard dimensiunea acestuia.