1. (0,75p) Scrieți o funcție care desparte un cuvânt în limba română în silabe, aplicând regulile de la <http://www.limba-romana.net/lectie/Reguli-de-despartire-a-cuvintelor-in-silabe/72/>
2. (0,50p) Scrieți o funcție care primește la intrare un șir de caractere folosind taguri HTML pentru **bold**, *italic* și underline și le elimină.
3. (0,75p) Scrieți o funcție care codifică un număr natural între 0 și 100000000 într-un șir de caractere, reprezentând numărul scris cu litere, în limba română. De exemplu, pentru numărul 70054 rezultatul funcției va fi „saptezecidemiicincizecisipatru”.
4. (1p) Scrieți o funcție care codifică un număr natural între 0 și 100000 într-un șir de caractere, reprezentând numărul scris cu litere, în limba franceză. De exemplu, pentru 74392 rezultatul funcției va fi „soixante-quatorze mille trois cent quatre-vingt-douze”. Pentru reguli, puteți consulta <http://www.heartandcoeur.com/convert/convert_chiffre_lettre.php>
5. (0,50p) Scrieți o funcție care primește la intrare un text sub forma unui șir de caractere și returnează numărul de cuvinte din text. Se consideră cuvânt orice secvență de litere (mici sau mari), cifre sau simbolul cratimă (reprezentat prin caracterul „-” (minus)), restul caracterelor considerându-se separatori.
6. (0,75p) Se dă un text aliniat la stânga, care cuprinde cuvinte (compuse doar din litere mici/mari, simbolul „-” (cratimă=minus) și simbolurile „ ” (spațiu) sau semnele de punctuație din limba română). Să se alinieze acest text la dreapta, eliminând eventualele spații duble dintre cuvinte.
7. (0.50p) Pentru a reprezenta amplasarea unei piese pe o tablă de șah se consideră o codificare printr-un șir de 4 caractere. Primul caracter reprezintă culoarea piesei, al doilea tipul piesei, a treia linia, iar a patra coloana. Scrieți o funcție care verifică dacă o anumită piesă atacă sau nu o altă piesă.
8. (0.80p) Scrieți o funcție care primește
   1. un vector de *n* numere a căror reprezentare binară reprezintă mulțimi de numere din intervalul [0,63]. De exemplu, codificarea binară 00100011 (pe 8 biți), corespunzătoare numărului zecimal 35, reprezintă mulțimea {0,1,5} (biții setați pe 1).
   2. un vector de n-1 caractere, fiecare caracter reprezentând o operatie între mulțimi: ’U’ = reuniune, ’A’ = intersecție, ’\’ = A – B, ’/’ = B – Ași returnează numărul obținut prin aplicarea operațiilor pe numerele primite ca parametru astfel:
      * Prima operație se aplică pe primele două numere
      * A doua operație se aplică pe rezultatul operației anterioare și al treilea numar
      * A treia operație se aplică pe rezultatul operației anteriore și al patrulea număr
      * ș.a.md.

Ex: Pentru sets=[1,2,3] și operations=[’U’,’\’] funcția va calcula 001(1) reunit cu 010(2) și va avea rezultatul 011 iar 011 minus 011(3) va avea rezultatul 000(0) si funcția va returna 0.  
*unsigned long setOperations(long sets[], char operations[], unsigned int x);*

1. (0.4p) Scrieți o funcție care pentru un vector de lungime *n* verifică dacă    elementele din vector sunt primele *n* numere din șirul lui Fibonnaci în     ordinea apariției lor in șir.  
   Ex: Pentru [0,1,1,2,3,4] funcția va returna true;  
   Pentru [0,1,1,3,2,4] si [0,1,1,2,4] funcția va returna false.  
   *bool areOrderedFibonnaci(vector);*
2. (0.4p) Scrieți o funcție care primește doi vectori ca parametru și returnează 0 dacă sunt egali, 1 daca primul este inclus în al doilea, 2 dacă al doilea este inclus în primul și 3 altfel.  
   Ex: Pentru [1,2,3] si [3,1,2] funcția returnează 0;  
   Pentru [1,2,3] si [3,2] funcția returnează 2;  
   Pentru [1,2,3] si [3,2,4] funcția returnează 3.  
   *unsigned char checkVectorInclude(vector vecOne, vector vecTwo);*
3. (0.5p) Scrieți o funcție care pentru un vector și o matrice primite ca parametri verifică dacă vectorul se regasește ca linie și/sau coloană în matrice.  
   *bool checkIsIn(vector vec, matrix mat);*
4. (0.5p) Scrieți o funcție care primește ca parametri o matrice patratică *n\*n*, un număr de rotații stânga și un număr de rotații dreapta și returnează matricea cu rotații stânga, respectiv dreapta în funcție de numerele primite ca parametri.  
   *matrix rotate(matrix mat, unsigned int rotLeft, unsigned int rotRight);*
5. (0.5p) Scrieți o funcție care pentru un vector de  lungime x și un număr y verifică dacă elementele din vector sunt primele x numere din sirul lui Fibonnaci începând cu numărul y (>=2).  
   Ex: Pentru [3,5,2] și 2 functia va returna true;  
   Pentru [3,4,2] si 2 functia va returna false.  
   *bool isPartOfFibonnaci(vector vec, unsigned int startingNumber);*
6. (0.6p) Scrieți o funcție care primește ca parametri un vector de x numere  și un vector de x-1 caractere reprezentând operații pe biți () și returnează rezultatul aplicării operațiilor pe numere. Operațiile sunt aplicate astfel: prima operație se aplică pe primele două numere, a doua operație se aplică pe rezultatul primei operații și al treilea număr, a treia operație pe rezultatul operației a doua și al patrulea număr ș.a.m.d.  
   *unsigned long bitOperations(long numbers[], char operations[], unsigned int x);*
7. (0.85p) Scrieți o funcție care verifică dacă reprezentarea binară a unui număr dat ca parametru este palindrom.  
   *bool palindrom(long number);*
8. (0.85p) Scrieți o funcție care primește ca parametru o matrice cu x rânduri și y coloane și verifică daca matricea conține primele (x\*y) elemente din sirul lui fibonnaci in spirală pornind din colțul stânga sus.  
   Ex: Pentru [0,1] [2,1] și [0, 1, 1] [13,21,2] [8, 5, 3] returnează true; Pentru [0,1] [1,2] returnează false.  
   *bool fibonnaciSpirale(matrix);*
9. (0,5p) Scrieți o funcție care primește o matrice de 1 și 0, și returnează matricea modificată astfel: dacă într-o celulă se găsește 0, toată linia și coloana respectivă vor fi umplute cu 0. Ex: Pentru matricea [1, 0, 1] [1, 1, 1] [1, 1, 1] funcția va returna [0, 0, 0] [1, 0, 1] [1, 0, 1]  
   *void transformMatrix(matrix mat);*
10. (0.25p) Să se elimine dintr-o listă de numere reale acele numere care au partea zecimală egală cu zero.
11. (0.25p) Suma elementelor de sub diagonala, de pe o linie, cu exceptia unei linii/coloane etc.
12. (0.20p). Se citește un vector *x* cu *n* numere întregi. Să se elimine din *x*elementele negative, în timp O(n).
13. (0.25p) Se citește un vector x cu n numere întregi. Sa se reașeze elementele din *x*astfel încât mai întâi să apară elementele negative, apoi cele pozitive, păstrând ordinea atât a elementelor negative, cât și a celor pozitive (O(n)).
14. (0,4p) Se citește un vector *a* cu *n* numere întregi, *n<1001*. Să se calculeze *s* = suma maximă care poate fi obținută dintr-o secvență de elemente de pe poziții consecutive din vector.  
    Exemplu: n=9, a=[-2,1,-3,4,-1,2,1,-5,4]. Rezultatul va fi s=6.
15. (1p) Scrieți o funcție care primește un labirint sub forma unei matrici, poziția celulei de plecare și poziția celulei de iesire și returnează lungimea drumului minim de la plecare la ieșire. Fiecare celulă din matrice va avea va avea valoare 1 (perete) sau 0 (drum).  
    *unsigned int minRouteLength(maze magicMaze);*
16. (0.25p fiecare) Exercitii cu siruri de caractere (de exemplu rescrierea funcțiilor uzuale din <string.h>)