Tema 1
Introducere în Programare
2016 - 2017

1.(0.4p) Scrieţi o funcţie care verifică dacă un număr este palindrom. Un număr este palindrom dacă citit de la dreapta la stanga este egal cu numărul citit de la stânga la dreapta.

Ex: 13231 este palindrom dar 12331 nu este palindrom

bool isPalindrom(unsigned long long number);

2.(0.4p) Scrieţi o funcţie care, pentru un număr natural, calculeză suma cifrelor reprezentării lui binare.

Ex: pentru number = 11 reprezentarea binară: 1011
functia va returna 3 (numarul 3 nu caracterul '3')

unsigned char sumBinaryFigure(unsigned long long number);

3.(0.4p) Scrieti o funcţie care calculează dacă un an este bisect.

bool isLeapYear(unsigned short year);

4.(0.4p) Scrieţi o funcţie care pentru o zi (an luna zi) va returna numărul zilei din săptamană.

Ex: pentru year = 2016 month = 10 (Octombrie) day = 1
funcţia va returna 6 (Sâmbătă) (numărul 6 nu caracterul '6')

Hint: Congruenţa lui Zeller

```
unsigned char dayOfTheWeek(unsigned short year,
                           unsigned char month,
                           unsigned char day);
```

5.(0.4p) Scrieti o funcţie care pentru un număr X returnează al X-lea număr din şirul lui Fibonnaci.

Şirul lui Fibonnaci:
fib(0) = 0;
fib(1) = 1;
fib(x) = fib(x-1) + fib(x-2);

Ex: fib(3) = fib(2) + fib(1) = fib(1) + fib(0) + 1 = 1 + 0 + 1 = 2

unsigned int fibonnaci(int index);

6.(0.4p) Spunem că un număr natural este perfect dacă este egal cu suma divizorilor săi strict mai mici decât el. Scrieţi o funcţie care returnează suma celor mai mari două numere perfecte mai mici decat un număr (number >= 30) dat.

Ex: 6 = 1 + 2 + 3 este numar perfect
pentru number = 30 numerele perfecte sunt 6 si 28

unsigned long perfectNumbers(unsigned int number);

7.(0.4p) Scrieţi o funcţie care pentru un interval închis dat [left, right] returnează numărul de numere ce au număr maxim de divizori primi.

Ex: Pentru [30, 45] numerele cu număr maxim de divizori primi = 30,42 (au cate 3 divizori primi, nici un alt număr din interval nu are mai mulţi divizori primi) iar functia va returna 2.

unsigned short primeDivisors(unsigned int left, unsigned int right);

8.(0.4p) Două numere naturale impare consecutive și prime se numesc numere prime gemene.Scrieți o funcție care determină primele x perechi de numere prime gemene mai mari decât un y dat.

Ex: Pentru count = 1 și lowerBound = 2 funcția va returna o matrice cu un rând (un rezultat) și 2 coloane (pereche de numere – două numere) 3 5

matrix primeTwins(unsigned int count, unsigned int lowerBound);

9.(0.4p) Scrieți o funcție care pentru un vector de lungime x verifică dacă elementele din vector sunt primele x numere din șirul lui Fibonnaci în ordinea apariției lor in șir.

Ex: Pentru [0,1,1,2,3,5] funcția va returna true;
Pentru [0,1,1,3,2,4] si [0,1,1,2,4] funcția va returna false.

bool areOrderedFibonnaci(vector vec);

10.(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 [1,2,1] si [3,1,2,1,4] funcția returnează 1;
Pentru [1,2,3] si [3,2,4] funcția returnează 3.

unsigned char checkVectorInclude(vector vecOne, vector vecTwo);

11.(0.4p) 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.

Ex: Pentru [1,3,6] și
[1,2,3]
[3,4,5]
[6,7,8]
funcția returnează true;
Pentru [1,2] și
[2,1]
[3,4] funcția returnează false.

bool checkIsIn(vector vec, matrix mat);

12.(0.4p) Scrieți o funcție care primește ca parametri o matrice patratică nxn, 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.

Ex: Pentru
[1,2]
[3,4],
rotLeft = 1, rotRight = 0 funcția returnează
[2,4]
[1,3].
Pentru
[1,2]
[3,4],
rotLeft = 0, rotRight = 1 funcția returnează
[3,1]
[4,2]

Pentru

[1,2,3]
[4,5,6]
[7,8,9],

rotLeft = 0, rotRight = 1 functia returneaza

[7,4,1]
[8,5,2]
[9,6,3]

matrix rotate(matrix mat, unsigned int rotLeft, unsigned int rotRight);

13.(0.6p) 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);

14.(0.6p) Scrieți o funcție care primește

- un vector de x numere a căror reprezentare binară reprezintă mulțimi de numere din intervalul [0,31]
- un vector de x-1 caractere, fiecare caracter reprezentând o operatie între mulțimi: 'U' = reuniune, 'A' = intersecție, '\' = A – B, '/' = B – A

și 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 anterioare 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);

15.(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.

Operații: < (<<), > (>>), ^, |, &

Ex: Pentru numbers=[1,2,3] și operations=[>,&] funcția va returna (1>>2)&3 = 0

unsigned long bitOperations(long numbers[], char operations[], unsigned int x);

16.(0.85p) Scrieți o funcție care verifică dacă reprezentarea binară a unui număr dat ca parametru este palindrom.

Ex: pentru number = 3 functia va returna false

bool palindrom(long number);

17.(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);

18.(0.85p) 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).

Ex: Pentru un maze
[0,0]
[0,0]
punct de start (0,1) si punct de finish (1,0)
un drum de lungime minima este (0,1) -> (0,0) -> (1,0)
si functia va returna 2.

unsigned int minRouteLength(maze magicMaze);

19.(0.85p) Scrieți o funcție care primește o matrice de 1 și 0, și modifică matricea 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 modifica matricea în
[0, 0, 0]
[1, 0, 1]
[1, 0, 1]

void transformMatrix(matrix mat);