

Probleme proiect TP 2019

1. BITPERM

Implementați un algoritm care citește de la intrarea standard două numere naturale și scrie la ieșirea standard dacă reprezentarea binară a unuia dintre numere poate fi obținută prin permutarea cifrelor din reprezentarea binară a celuilalt.

Exemplul 1

Intrare	Ieșire
13 7	DA

Explicație: cum numărului 13 îi corespunde valoarea 1101_2 iar numărului 7 valoarea 0111_2 se va scrie la ieșirea standard "DA" pentru că putem obține 0111_2 prin permutarea cifrelor lui 1101_2 .

Exemplul 2

Intrare	Ieșire
8 7	NU

Explicație: cum numărului 8 îi corespunde valoarea 1000_2 iar numărului 7 valoarea 0111_2 se va scrie la ieșirea standard "NU" pentru că nu putem obține 0111_2 prin permutarea cifrelor lui 1000_2 și nici 1000_2 prin permutarea cifrelor lui 0111_2 .

2. FINDODD

Implementați un algoritm care citește de la intrarea standard o secvență de numere naturale cuprinse între 0 și 9999 (inclusiv 0 și 9999) și scrie la ieșirea standard acele numere care apar de un număr impar de ori în secvența de intrare.

Exemplul 1

Intrare	Ieșire
1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	1 3 5

Explicație: cele trei numere care apar de un număr impar de ori în secvența de intrare sunt 1 (o singură apariție), 3 (trei apariții) și 5 (cinci apariții).

Exemplul 2

Intrare	Ieșire
1 2 3 4 5 4 3 2 1	5

Explicație: singurul număr care apare de un număr impar de ori este 5 (o apariție).

3. SORT

Se dă un vector cu N elemente, numere naturale, și un număr k. Ordonăți crescător primele K elemente ale vectorului și descrescător ultimele N-K elemente.

4. BITI

Se dă un număr natural N pe 32 de biți. Se cere să se calculeze numărul obținut prin interschimbarea biților de rang par cu cel de rang impar.

Exemplu

Intrare	Ieșire
2863311530	1431655765

5. CIFRE

Sa se determine numărul de cifre de 1 din reprezentarea binară a unui număr natural nenul n mai mic ca 2000000000.

6. GRAYCODE

Implementați un algoritm care citește de la intrarea standard un număr N ($1 \leq N \leq 32$) și scrie la ieșirea standard un cod Gray pe N biți.

Exemplul 1

Intrare	Ieșire
3	0 1 3 2 6 7 5 4
<i>Explicație:</i> valorilor obținute corespund reprezentările binare 000_2 , 001_2 , 011_2 , 010_2 , 110_2 , 111_2 , 101_2 și respectiv 100_2 . Cum între reprezentările binare ale oricăror două valori consecutive diferă o singură cifră acesta este un cod Gray pe trei biți.	

7. PARANOK

Implementați un algoritm care citește de la intrarea standard o expresie matematică și scrie la ieșirea standard dacă expresia este corect parantezată.

Exemplul 1

Intrare	Ieșire
$1+(20*3-(-1))$	DA
<i>Explicație:</i> fiecărei paranteze deschise îi corespunde o paranteză închisă.	

Exemplul 2

Intrare	Ieșire
$1+(20*3-(-1)$	NU
<i>Explicație:</i> primei paranteze deschise nu îi corespunde o paranteză închisă.	

Exemplul 3

Intrare	Ieșire
$1+20*3-(((-1)))$	DA
<i>Explicație:</i> fiecărei paranteze deschise îi corespunde o paranteză închisă.	

Exemplul 4

Intrare	Ieșire
$1+20*3-(-1\}$	NU
<i>Explicație:</i> primei paranteze deschise nu îi corespunde o paranteză închisă iar primei paranteze închise nu îi corespunde o paranteză deschisă.	

8. PARACST

Implementați un algoritm care citește de la intrarea standard o expresie matematică și scrie la ieșirea standard costul acestei expresii. Definim în acest sens costul unei expresii ca suma costurilor perechilor de paranteze care o compun iar costul unei perechi de paranteze ca diferența dintre poziția pe care apare paranteza închisă și cea pe care apare paranteza deschisă corespunzătoare ei.

Exemplul 1

Intrare	Ieșire
1+(2)	2

Explicație: există o singură pereche de paranteze (cea cu paranteza închisă pe poziția 5 iar paranteza deschisă pe poziția 3) iar costul acestei perechi este 2.

Exemplul 2

Intrare	Ieșire
1+(2000)	5

Explicație: există o singură pereche de paranteze (cea cu paranteza închisă pe poziția 8 iar paranteza deschisă pe poziția 3) iar costul acestei perechi este 5.

Exemplul 3

Intrare	Ieșire
1+(2)*([3])	8

Explicație: există trei perechi de paranteze:

- cea cu paranteza închisă pe poziția 5 iar paranteza deschisă pe poziția 3 (cost 2);
- cea cu paranteza închisă pe poziția 11 iar paranteza deschisă pe poziția 7 (cost 4);
- cea cu paranteza închisă pe poziția 10 iar paranteza deschisă pe poziția 8 (cost 2).

Suma costurilor perechilor de paranteze, 8, reprezintă costul expresiei.

9. EVALLOG

Implementați un algoritm care citește de la intrarea standard o expresie logică și scrie la ieșirea standard rezultatul evaluării acestei expresii. Expresia logică de evaluat conține exclusiv

constantele T (*TRUE*) și F (*FALSE*), operatul unar ! (*NOT*), operatorii binari * (*AND*), + (*OR*) și ^ (*XOR*) și paranteze rotunde.

Exemplul 1

Intrare	Ieșire
T+F	T

Exemplul 2

Intrare	Ieșire
!!T*!!!(F)	T

10. PERMDIF

Implementați un algoritm care citește de la intrarea standard două numere N și D și scrie la ieșirea standard, în ordine lexicografică, acele permutări ale primelor N numere naturale pentru care diferența în modul între oricare două elemente consecutive este mai mică sau egală cu D.

Exemplul 1

Intrare	Ieșire
3 1	0 1 2 2 1 0

Explicație: dintre cele șase permutări ale primelor trei numere naturale, {0, 1, 2}, {0, 2, 1}, {1, 0, 2}, {1, 2, 0}, {2, 0, 1} și {2, 1, 0}, permutările {0, 1, 2} și {2, 1, 0} sunt singurele pentru care diferența în modul între oricare două elemente consecutive este mai mică sau egală cu 1.

Exemplul 2

Intrare	Ieșire
3 2	0 1 2 0 2 1 1 0 2 1 2 0 2 0 1 2 1 0

Explicație: toate cele șase permutări ale primelor trei numere naturale, {0, 1, 2}, {0, 2, 1}, {1, 0, 2}, {1, 2, 0}, {2, 0, 1} și {2, 1, 0} au diferența în modul între oricare două elemente consecutive mai mică sau egală cu 2.

11. RSC

Se dau n obiecte transportabile cu ajutorul unui rucsac. Pentru fiecare obiect i se cunoaște greutatea $g(i)$ și costul $c(i)$. De asemenea se cunoaște greutatea G pe care o poate suporta rucsacul. Se dorește încărcarea rucsacului cu obiecte astfel încât suma greutăților obiectelor alese să fie mai mic sau egal cu G , iar suma costurilor obiectelor alese să fie maxim. Să se afișeze pe prima linie greutatea și costul maxim obținute prin alegerea unei submultimi din mulțimea de obiecte dată. Pe următoarea linie să se afișeze numărul de ordine al obiectelor alese și bagate în rucsac.

Exemplul 1

Intrare	Ieșire
4 5	4 9
3 3	1 2
1 5	
3 4	
5 8	

Explicație:

4 obiecte, G rucsac este 5. Se aleg obiectele 1 și 2 (numerotarea începe de la 0) care au greutatea însumată 4 și costul 9.

12. LAB

Se citește de la intrare o matrice binară de dimensiune $n \times m$, reprezentând un labirint, $A[i][j] = \{0, 1\}$, $0 \leq i < n$, $0 \leq j < m$. $A[i][j] = 1$ – la punctul de coordonate i și j se află un zid, $A[i][j] = 0$ – la punctul de coordonate i și j nu există zid. Se dă un punct inițial $P(x_1, y_1)$ și un punct final $Q(x_2, y_2)$. Știind că prin labirint se pot face deplasări sus, jos, stânga, dreapta și că zidurile pot fi sparte, se dorește găsirea unui drum între P și Q cu proprietatea că numărul de ziduri sparte să fie minim.

Să se afișeze la ieșire numărul de ziduri sparte și drumul prin coordonate (x, y) . Dacă sunt mai multe drumuri se va afișa oricare dintre ele.

Exemplul 1

Intrare	Ieșire
6 7	1
1 1	(1,1)(1,2)(0,2)(0,3)(0,4)(0,5)
4 2	(1,5)(2,5)(2,4)(3,4)(4,4)(4,3)(4,2)
0 0 0 1 0 0 1	
0 0 0 1 1 0 1	
1 1 1 1 0 0 1	
1 1 1 1 0 1 1	
0 0 0 0 0 0 1	
0 0 0 1 0 0 1	

Explicație: P(1,1) și Q(4,2). Se sparge un singur zid

13. PAY

Implementați un algoritm care citește de la intrarea standard un număr S reprezentând o sumă de achitat și o secvență M reprezentând tipurile de monede disponibile și scrie la ieșirea standard, în ordine lexicografică, toate modalitățile distincte de achitare a sumei S folosind oricâte monede de oricare dintre tipurile disponibile.

Observație: două modalități de achitare a unei sume sunt considerate una și aceeași (nu sunt considerate distincte) dacă una dintre modalități reprezintă o permutare a celeilalte.

Exemplul 1

Intrare	Ieșire
4	1 1 1 1
1 2 5	1 1 2
	2 2

Explicație: putem achita suma de 4 cu patru monede de 1 (1 1 1 1), cu două monede de 1 și o monedă de 2 (1 1 2) sau cu două monede de 2 (2 2).

Exemplul 2

Intrare	Ieșire
5	1 1 1 1 1
1 2 3	1 1 1 2
	1 1 3
	1 2 2
	2 3

Explicație: putem achita suma de 5 cu cinci monede de 1 (1 1 1 1 1), cu trei monede de 1 și o monedă de 2 (1 1 1 2), cu două monede de 1 și o monedă de 3 (1 1 3), cu o monedă de 1 și două de 2 (1 2 2) sau cu o monedă de 2 și o monedă de 3 (2 3).

14. LCPN

Se da un sir de cifre. Sa se afiseze toate combinatiile de litere conform tastelor telefonului

Intrare	Ieșire
26	am an ao bm bn bo cm cn co

15. FBK

Pentru multimea de numere $\{x_1, x_2, x_3, x_4, \dots, x_N\}$ ($N \geq 3$) multimea sumelor de perechi este $\{x_1+x_2, x_1+x_3, x_1+x_4, x_2+x_3, x_2+x_4, x_3+x_4, \dots\}$. ($s_k = x_i + x_j, i \neq j$). Se da multimea sumelor de perechi si se cere gasirea multimii initiale. Multimea sumelor de perechi se da intr-o ordine oarecare.

Intrare	Ieșire
6 105 11 101 110 15	1 5 10 100

16. INV

Se citesc din fisierul de intrare in.txt N numere naturale reprezentand permutarea unei multimi cu N elemente. Sa se afiesze numarul de inversiuni din permutarea data. Complexitate maxima $O(n \log n)$. Pentru punctaj maxim se vor citi mai multe linii din fisierul de intrare.

Intrare	Ieșire
3 1 2	2
1 2 3 4	0
2 1	1

Explicație: permutarea (3 1 2) contine 2 inversiuni: [3 1] și [3 2]

17. MAT

Se da o matrice de dimensiune $N \times N$. Sa se roteasca continutul matricii cu 90 de grade la dreapta fara a se utiliza vectori sau matrici auxiliare.

Intrare	Ieșire
3	7 4 1
1 2 3	2 5 2
4 5 6	2 6 3
7 2 2	

18. PTR

Sa se calculeze n la puterea k . Complexitate $O(\log k)$.

Intrare	Ieșire
2 3	8
3 4	81

19. Mod

Sa se calculeze n la puterea k modul p . $n, k, p \leq 1.000.000.000$. Complexitate $O(\log k)$.

Intrare	Ieșire
9 14 7	4
13 11 99999	15820

Explicație: $9^{14} \% 7 = 4$, $13^{11} \% 99999 = 15820$

20. Bta

Se dau doua numere naturale N ($1 \leq N \leq 10000$) si K ($1 \leq K \leq 100$), apoi N numere naturale scrise in fisierul de intrare. Dintre valorile acestora, toate mai putin una apare de un numar de ori multiplu de K . Sa se gaseasca si sa se afiseze valoarea numarului care nu respecta aceasta proprietate.

Intrare	Ieșire
17 3 3 4 4 4 3 6 7 7 7 3 6 3 3 4 3 4 4	6

21. Lex

Se da un numar T ($1 \leq T \leq 1000$), reprezentand numarul de teste. Pe fiecare dintre urmatoarele T randuri se afla cate un sir format doar din valorile 0 si 1. Pentru fiecare sir s de lungime L ($1 \leq L \leq 32$) sa se raspunda la intrebarea: al catelea sir in ordine lexicografica este, daca consideram toate sirurile de lungime L , formate doar din valorile 0 si 1.

Intrare	Ieșire
4	2
1	6
101	1
0	63
111110	

22. Drt

Se dau urmatoarele operatii:

1. Creare patrat de dimensiuni 1×1 .
2. Extindere patrat deja existent, cu o unitate. De exemplu, daca aplicam operatia de extindere pe un patrat de dimensiuni $A \times A$, dupa aplicare, patratul va avea dimensiuni $(A + 1) \times (A + 1)$.

Citindu-se N si M , numere naturale, afisati numarul minim de operatii pentru a construi un dreptunghi de dimensiuni $N \times M$.

Intrare	Ieșire
7 4	10

23. Ale

Parcul orasului a fost neglijat mult timp, astfel ca acum toate aleile sunt distruse. Prin urmare, anul acesta Primaria si-a propus sa faca reamenajari. Parcul are forma unui patrat cu latura de N metri si este inconjurat de un gard care are exact doua porti. Proiectantii de la Primarie au realizat o harta a parcului si au trasat pe harta un caroi aj care imparte parcul in $N \times N$ zone patrute cu latura de 1 metru. Astfel harta parcului are aspectul unei matrice patrute cu N linii si N coloane. Liniile si, respectiv, coloanele sunt numerotate de la 1 la N . Elementele matricei corespund zonelor patrute de latura 1 metru. O astfel de zona poate sa contina un copac sau este libera. Ediliile orasului doresc sa paveze cu un numar minim de dale patrute cu latura de 1 metru zonele libere (fara copaci) ale parcului, astfel incat sa se obtina o alee continua de la o poarta la alta.

Scrieti un program care sa determine numarul minim de dale necesare pentru construirea unei alei continue de la o poarta la cealalta.

Fisierul de intrare contine pe prima linie doua valori naturale N si M separate printr-un spatiu, reprezentand dimensiunea parcului, respectiv numarul de copaci care se gasesc in parc. Fiecare dintre urmatoarele M linii contine cate doua numere naturale X si Y separate printr-un spatiu, reprezentand pozitiile copacilor in parc (X reprezinta linia, iar Y reprezinta coloana zonei in care se afla copacul). Ultima linie a fisierului contine patru numere naturale X_1, Y_1, X_2, Y_2 , separate prin cate un spatiu, reprezentand pozitiile celor doua porti (X_1, Y_1 reprezinta linia si respectiv coloana zonei ce contine prima poarta, iar X_2, Y_2 reprezinta linia si respectiv coloana zonei ce contine cea de a doua poarta).

Fisierul de iesire va contine o singura linie pe care va fi scris un numar natural care reprezinta numarul minim de dale necesare pentru construirea aleii.

Intrare	Ieșire
8 6 2 7 3 3 4 6 5 4 7 3 7 5 1 1 8 8	15

24. Flr

Lalelele din Parcul Botanic au fost numerotate de la 1 la n ($n \leq 100$). Se dorește formarea unui buchet, care să conțină cel puțin o floare, iar două flori numerotate consecutiv să nu aparțină buchetului.

Fiind dat n , numărul de flori, să se determine în câte moduri se poate forma buchetul.

Fișierul de intrare conține pe prima linie un număr natural n , reprezentând numărul de flori.
Fișierul de ieșire conține pe prima linie un număr natural ce reprezintă numărul de buchete modulo 9001.

Intrare	Ieșire
5	12

Observații:

1. Toate datele de intrare se citesc din fișierul **in.txt**, iar rezultatele se afișează în **out.txt**, dacă nu se specifică altfel în textul problemei
2. Codul se predă **identat și comentat**
3. Orice întrebare legată de cerința unei probleme se pune la curs/laborator
4. Fiecare problemă va fi punctată cu un punctaj ponderat în funcție de numărul rezolvărilor corecte. Media pe proiect se calculează ca medie aritmetică între punctajele obținute pe fiecare problemă. Exemplu: $\text{MIN}((7+8+17)/3, 10)=10$; pe problema trei au fost acordate 17 puncte pentru că a fost una dintre puținele rezolvări corecte (ponderare)
5. Verificarea fiecărei probleme se va face cu mai multe fișiere de intrare/ieșire
6. Codul se scrie exclusiv în **ANSI C (sursele c++ nu vor fi luate în considerare)**
7. Fișierele pt. fiecare problemă se salvează în felul următor: **1_NumePrenume.c**, **15_NumePrenume.c**, **i_NumePrenume.c**, unde i reprezintă numărul problemei
8. Sursele se salvează în directorul **NumePrenume**, apoi se împachetează (format **zip**, orice alt format nu va fi luat în considerare) într-un fișier **NumePrenume.zip** și se trimit până luni **12.05 ora 24**, pe adresa de email proiect.tp@gmail.com. Subiectul mesajului va fi format din numele și prenumele studentului separate prin câte un spațiu.
Ex:
BanasOvidiu\1_BanasOvidiu.c
BanasOvidiu\3_BanasOvidiu.c
BanasOvidiu\4_BanasOvidiu.c

->BaniasOvidiu.zip

8. Se trimit maxim 4 fișiere (doar sursele)
9. Sursele trimise după această dată/oră nu vor fi luate în considerare
10. Punctajul fiecărei probleme va fi în intervalul $[2,10]$
11. Nepredarea proiectului implică automat nota 2
12. Nota obținută la proiect nu poate fi mărită, marire se poate da doar la testul grilă
13. Nota finală la examen se va calcula ca medie aritmetică între nota la proiect și nota la testul grilă