

PROBLEME - PROIECT TP 2020

1. RLE

Fiind dat un șir de caractere (litere mici ale alfabetului englez) de lungime N , să se codifice șirul după următoarea regulă: fiecare secvență maximală formată din același caracter va fi înlocuită cu caracterul urmat de lungimea secvenței. **Complexitate maximă $O(N)$.**

Input	Output
aaacccccbbdee	a3c5b2d1e2

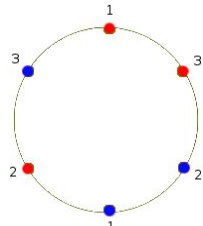
2. ATM

Fie un bancomat ce conține 8 tipuri de bancnote: 1, 5, 10, 20, 50, 100, 200 și 500. Numărul de bancnote de fiecare tip este dat ca fiind primele opt valori de intrare. În decursul unei zile vin N clienți ($0 < N \leq 1\,000$) să extragă bani din bancomat. Să se afișeze dacă se pot oferi exact sumele de bani pentru toți clienții. Sumele cerute V ($V \leq 10\,000\,000$) de clienți vor fi date ca un vector de N numere naturale. **Complexitate maximă $O(N \times V)$.**

atm.in	atm.out
10 5 8 4 6 20 15 7 6 20 45 300 70 125 600	DA
10 5 8 4 6 20 15 7 6 20 45 30000 70 125 600	NU

3. SRV

Într-un oraș circular, numărul total de locuitori este N . Fiecare dintre aceștia are o casă și un birou propriu, deci în total există $2N$ construcții, așezate la periferia orașului. Distanța între oricare două clădiri consecutive este egală cu 1. Dându-se cele N perechi de numere care reprezintă pozițiile casei și a biroului fiecărui locuitor, să se afișeze distanța maximă parcursă de un locuitor, știind că fiecare locuitor alege mereu drumul cel mai scurt de la casa sa spre birou. **Complexitate maximă $O(N)$.**

srv.in	srv.out	Explicații
3 1 4 5 3 2 6	3	Casele sunt cu roșu, birourile cu albastru. 

4. CUV

Se dau lungimile a **N** cuvinte ($0 < N \leq 5\,000$), formate din cel puțin un caracter. Să se afișeze lungimea minimă necesară **R** a unui rând dintr-o pagină, astfel încât pe fiecare rând lungimea cumulată a cuvintelor să fie exact R (fără spații între ele). Cuvintele nu au voie să fie despărțite.

cuv.in	cuv.out
11 10 2 8 3 1 3 9 6 6 4 8	12

5. SWP

Se dă un șir de **N** biți ($0 < N < 1\,000$). Acest șir trebuie sortat crescător folosind doar interschimbări ale pozițiilor. Să se afișeze numărul minim de interschimbări necesare, precum și toate interschimbările efectuate. Orice soluție minimă e acceptată. **Complexitate maximă $O(N)$.**

swp.in	swp.out	Explicații
8 10101010	2 1 6 3 8	Pozițiile se numerează de la stânga la dreapta, începând cu poziția 1. După aplicarea sortării, șirul va deveni 00001111 .

6. JUN

Se dă un număr natural **N**. Să se afișeze în câte moduri distincte poate fi scris acest număr ca sumă de două numere naturale pare nenule. Două moduri sunt distincte dacă unul dintre ele **nu** este o comutativitate a celuilalt mod. **Complexitate maximă $O(\log N)$.**

7. PRA

Se dau **N** numere naturale ($0 < N \leq 10$). Să se afișeze câte subșiruri crescătoare formate din cel puțin doi termeni reprezintă o progresie aritmetică și care sunt acestea. Subșirurile pot fi afișate în orice ordine, câte un subșir pe o linie.

pra.in	pra.out	Explicație
4 2 3 4 6	8 2 3 2 3 4 2 4	Totale subșirurile sunt: (2 3) (2 3 4) (2 4) (2 4 6) (2 6) (3 4) (3 6) (4 6)

8. AMS

Se dau **N** cuvinte ($N \leq 10\,000$), de lungime maximă **L** caractere ($L \leq 1\,000$). Pentru fiecare cuvânt introdus, creați și afișați pe câte o linie un nou cuvânt format astfel încât să fie:

- 1) Format prin amestecarea literelor cuvântului original.
- 2) Mai mare din punct de vedere lexicografic decât cuvântul original.
- 3) Cel mai mic cuvânt din punct de vedere lexicografic respectând condițiile 1) și 2).

Dacă cel puțin una dintre cele trei condiții nu poate fi respectată, se va afișa -1 pentru cuvântul respectiv. **Complexitate maximă $O(N \times L)$ timp și $O(L)$ memorie.**

ams.in	ams.out	Explicații
4 lmno dcba abdc abcd	lmon -1 acbd abdc	Se citește numărul N, apoi cele N cuvinte, câte un cuvânt pe un rând. Se vor afișa doar cuvintele transformate, în ordinea datelor de intrare.

9. ZIG

Se consideră un tabel în care se află toate numerele de la 1 la 2^{2N} în următorul format:

N = 1

4	3
2	1

N = 2

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

Dându-se un număr natural nenul **N** ($N \leq 8$) și **Q** ($0 < Q \leq 1\,000$) perechi de valori întregi **X** și **Y**, să se afișeze care este valoarea din tabel la fiecare dintre coordonatele (X,Y). Coordonatele tabelului încep de la (1,1) și se garantează că sunt valide în matrice. De pe prima linie se citesc numerele N și Q, apoi urmează Q linii cu coordonate. **Complexitate maximă $O(Q \times N^2)$.**

zig.in	zig.out
2 3 3 4 2 1 1 4	3 14 11

10. XBG

Se dă un număr natural **N** pe 32 biți. Să se afișeze câte numerele naturale **Z** există ($Z < N$), care satisfac relația **(N xor Z) > N**. **Complexitate maximă $O(\log N)$.**

11. CMM

Se dau două numere naturale **A** și **B**, ($0 \leq A, B \leq 10^4$). Să se afișeze cel mai mare divizor comun și cel mai mic multiplu comun.

12. 102

Se citește un număr natural $N \leq 10^9$. Să se transforme **N** din baza 10 în baza 2 reținând cifrele într-un vector, apoi să se afișeze vectorul.

13. MAT

Se citesc **N** ($1 \leq N \leq 18$) și **K** ($1 \leq K \leq N$). Să se creeze o matrice cu **N** linii și **N** coloane care să conțină toate numerele de la 1 la N^2 astfel încât:

- Oricare două valori consecutive să fie alăturate în matrice (căsuțe alăturate pe orizontală sau verticală)
- Valoarea 1 să fie pe linia **K** (liniile se numerotează de sus în jos)

Se afișează elementele matricii.

Exemplu:

Input

4 'N=4

2 'K=2

Output

2 3 4 5

1 8 7 6

16 9 10 11

15 14 13 12

14. MUL

Se citesc două mulțimi de numere naturale **A** și **B**. Să se afișeze reuniunea, intersectia, diferența celor două mulțimi.

15. PAL

Se dau două numere **N** și **M**, ($0 \leq N \leq 20$, $0 \leq M \leq 10$). Să se afișeze toate palindroamele de lungime **M** cu elemente din mulțimea $\{1, \dots, N\}$.

Exemplu

Input

15 3

Output

1 2 1

....

9 15 9

....

16. CCO

Se da o tabletă de ciocolată de dimensiunea $M \times N$, ($0 \leq N \leq 100$, $0 \leq M \leq 100$) formată din cubulețe de forma 1×1 . Care este numărul minim de ruperi necesar pentru a separa complet toate cubulețele de 1×1 .

Exemplu

Input

2 2

Output

3

Tableta de 2×1 are nevoie de doar 1 rupere, rezultă două cubulețe de 1×1

Tableta de 2×2 are nevoie de 3 ruperi și rezultă 4 cubulețe de 1×1

17. PIO

Se dă o listă de poziții ale unor pioni de aceeași culoare pe o tabla de șah de 8×8 numerotată pe axa x de la "a" la "h" și pe axa y de la 1 la 8. Să se afle câți dintre acești pioni sunt "protejați". Conform regulilor de șah, un pion 'd4' este considerat protejat dacă există alt pion 'c3' sau 'e3'.

Exemplu

Input

b4 d4 f4 c3 e3 g5 d2

Output

6

{"b4", "d4", "f4", "c3", "e3", "g5", "d2"} -> 6 pioni protejati

{"b4", "c4", "d4", "e4", "f4", "g4", "e5"} -> 1 pion protejat

18. NBR

Se dă o listă de N ($1 \leq N \leq 10^9$) întregi și I ($1 \leq N \leq 10^9$) întrebări. Pentru fiecare index K ($0 \leq K < N$) din listă asociat fiecărei întrebări să se afișeze numărul de elemente ale secvenței maxime aflate la dreapta indexului dat K cu proprietatea că elementele secvenței sunt în ordine crescătoare. Timp maxim de execuție: 5 secunde.

Exemplu

Input

3 4 2 7 5 8 10 6

2 'l = 2

0 'index = 0

5 'index = 5

Output

4 '4 7 8 10 subsecvența maximală crescătoare de lungime 4

1 '10 sau 6 subsecvența maximală crescătoare de lungime 1

19. TRG

Începând din vârful triunghiului spre bază coborând doar pe elementele adiacente, să se calculeze suma maximă care poate fi obținută.

Pentru exemplul

```
  3
 7 4
2 4 6
8 5 9 3
```

ruta care ne-ar oferi suma maximă ar fi: 3 7 4 9 ($3 + 7 + 4 + 9 = 23$).

Se dă un triunghi de forma celui de mai sus. Să se calculeze suma maximă obținută din vârful triunghiului până la bază.

trg.in	trg.out
3 7 4 2 4 6 8 5 9 3	23

20. DUM

Să se determine cate "duminici" există între 2 date introduse de la tastatură considerând următoarele aspecte :

- 1 Ianuarie 1900 a fost o zi de luni,
- Un an bisect are loc o dată la 4 ani,
- 30 de zile au lunile Septembrie / Aprilie / Iunie / Noiembrie,
- Februarie are 28 de zile, 29 în an bisect,
- Restul lunilor au 31 de zile.

21. PRM

Numărul 197 este considerat un număr prim circular, deoarece orice combinație a cifrelor sale formează un alt număr prim.

Există 13 numere prime circulare sub 100: 2, 3, 5, 7, 11, 13, 17, 31, 37, 71, 73, 79, și 97.

Se citește **N** ($0 \leq N \leq 10^4$). Să se afișeze toate numerele prime circulare mai mici decât **N**.

22. PLA

Se citește **N** ($0 \leq N \leq 10^6$). Să se afișeze toate numerele mai mici decât **N** care sunt palindrome în cel puțin 2 baze de numerație. (Se considera baza 10,, baza 2, baza 8 și baza 16)

Numărul $585(10) = 1001001001(2)$, este palindrom atât în baza 10 cât și în baza 2.

23. COR

Se dă un string de cifre **S** ($1 \leq S \leq 100$), să se insereze un număr minim de paranteze rotunde deschise și închise astfel încât stringul să fie corect parantezat și fiecare cifră **C** să fie în exact **C** perechi de paranteze.

cor.in	cor.out
021	0((2)1)
312	((((3))1(2))
4	(((((4))))

24. BRI

O bară are lungimea **L** ($1 \leq L \leq 1000$). Se consideră **N** ($1 \leq N \leq 10$) repere de lungimi diferite. Să se genereze toate posibilitățile de a tăia bara după reperele existente, fără să rămână rest la tăiere, fiecare reper fiind folosit cel puțin o dată. Se citesc dintr-un fișier text, de pe primul rând, lungimea barei – **L** și numărul de repere – **n**, iar de pe următorul rând, reperele. Numerele de pe un rând sunt separate prin spațiu

bri.in	bri.out
13 3 7 1 2	1 1 1 1 2 7 1 1 2 2 7
15 4 1 2 3 4	1 1 1 1 1 1 2 3 4 1 1 1 1 2 2 3 4 1 1 1 2 3 3 4 1 2 2 3 3 4 1 1 2 2 2 3 4 1 1 2 3 4 4

Observații:

1. Toate datele de intrare se citesc din fisierul 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 trimite până luni 15.05 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
->BanasOvidiu.zip
8. Se trimite 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, mărire 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