Note:

1. Fiecare program va fi testat cu 5 teste a câte 10 puncte ⇒ **FIECARE PROBLEMA ARE MAXIMUM 50 PUNCTE**.
2. Fiecare program are numele Pr1...Pr5, salvate în mapa cu numele participantului.

**Clasa 10**

Problema 1. **Moneta falsa**

De la tastatură se introduc **n** numere – greutățile a **n** monede identice ca mărime dar diferite după greutate, dintre care una este falsă. De scris un program, care afișează la ecran a câta monedă este falsă și dacă e mai ușoară sau mai grea ca celelalte.

**Input:** Numărul **n** și **n** numere întregi se citeşte de la tastatură.

**Output:** Numărul de ordine a monedei false și una din frazele „mai grea”, „mai ușoară” se afişează la ecran

**Exemplu:** Input: **6 4 4 7 4 4 4**

Output: 3 **„mai grea”**

Problema 2. **Cifra**

De la tastatură se introduc 2 numere naturale ***a,b*** mai mici ca 30000 și o cifră ***c***. De scris un program, care afișează la ecran DA dacă cifra ***c*** se conține în suma ***a+b*** și NU în caz contrar

**Input:** Numerele ***a,b și c*** se introduc de la tastatură

**Output:** Se va afișa rezultatul la ecran.

**Exemplu:** **Input:** ***104 57 1***

**Output: DA**

Problema 3. **Tabele înrudite**

Se consideră tabloul A[1..N] de numere întregi, N<=100. Să se scrie un program care completează tabloul B[1..N div 2+1] în felul următor: primul element din B să fie egal cu suma primului şi ultimului element din A, elementul al doilea din B să fie egal cu suma dintre al doilea şi penultimul element din A. s. a. m. d.

**Input:** Fişierul TABELUL.A va conţine pe prima linie numărul natural N, și pe a doua linie N numere întregi despărțite prin spațiu

**Output:** Fişierul TABELUL.B va conţine o singură linie cu elementele tabloului B.

**Exemplu:** Input: **7**

**1 2 0 3 4 5 2**

Output: **3 7 4 3**

Problema 4. **Calcul**

Să se scrie un program care calculează valoarea ***2n*** (1≤ ***n*** ≤1000).

**Input:** Numărul ***n*** se introduce de la tastatură

**Output:** Rezultatul obținut se afişează la ecran.

**Exemplu:** **Input:** ***10***

**Output: *1024***

Problema 5. Cronometru

Ceasul electronic este un aparat care serveşte la determinarea timpului în limitele unei zile. Indicaţiile curente ** ale unui ceas electronic au formatul: hh:mm:ss, unde hh reprezintă ora ; mm – minutele ; ss – secundele .

Cronometrul electronic este un instrument pentru măsurarea intervalelor de timp . Elaboraţi un program care reprezintă indicaţiile cronometrului în formatul hh:mm:ss.

**Input:** Fişierul INPUT.TXT va conţine două linii cu indicaţiile curente respectiv .

**Output:** Fişierul OUTPUT.TXT va conţine o singură linie cu indicaţiile cronometrului .

## Exemplu:

|  |  |
| --- | --- |
| INPUT.TXT 12:06:15  16:51:02 | OUTPUT.TXT 04:44:47 |

Problema 6. **Cuvânt lung**

Se dă un text alcătuit din litere mici ale alfabetului latin şi cuvintele sunt separate printr-un spaţiu(lungimea textului≤255). Afişaţi la ecran cel mai lung cuvânt. Dacă astfel de cuvinte sunt câteva, de afişat primul din ele.

**Input:** **Input:** Textul se citeşte de la tastatură

**Output:** Cuvântul găsit se afişează la ecran

**Exemplu:** Input: **noroc are acela care vrea.**

Output: **noroc**

Problema 7. **Numere interesante**

Se ştie că de la numărul 4 se poate obţine orice număr natural scris în baza 10 prin aplicarea următoarelor operaţii:

1. Se scrie la sfârşit cifra 4;
2. Se adaugă la sfârşit cifra 0;
3. Se împarte la 2 (dacă numărul este par).

Să se scrie un program, care produce un şir de numere construit conform regulilor A – C, în care primul număr este 4 iar ultimul este N.

**Exemplu:** dacă N=15 calea este: 4 → C → 2 → A → 24 → C → 12 → C→ 6 → C → 3 → B→ 30 → C →15.

**Input:** Fişierul ***numere.txt*** va conţine pe unica linie - valoarea lui N

**Output:** Fişierul ***calea.txt*** va conţine calea obținută scrisă într-o linie fiind despărţite prin spaţiu. (Pentru exemplu 4 C 2 A 24 C 12 C 6 C 3 B 30 C 15).

Note:

1. Fiecare program va fi testat cu 5 teste a câte 10 puncte ⇒ **FIECARE PROBLEMA ARE MAXIMUM 50 PUNCTE**.
2. Fiecare program are numele Pr1...Pr5, salvate în mapa cu numele participantului.

**Clasele 11,12**

Problema 1. **Paranteze**

Se consideră o expresie aritmetică, unde sunt folosite parantezele (,{,[,],} şi ). Elaboraţi un program care verifică dacă ‚în expresie parantezele au fost folosite corect.

**Input:** Fişierul de intrare *paranteze.in* care conţine pe fiecare linie câte o expresie matematică.

**Output** Fişierul de ieşire *paranteze.out* va conţine pe fiecare linie cuvântul DA sau NU în funcţie de corectitudinea şirului respectiv din fişierul paranteze.txt.

**Exemplu:**

|  |  |
| --- | --- |
| Fişierul *paranteze.in* | Fişierul *paranteze.out*: |
| (a\*b)  (12-e)(d+a)[c( | DA  NU |

Problema 2. **Baza**

Baza numărului N este egală cu C+1, unde C este cifra maximă din numărul în studiu. Elaboraţi un program care transcrie numerele naturale din baza C+1 în baza 10.

**Input:** Fişierul de intrare input.txt care conţine pe fiecare linie câte un număr (scris în baza maximă 36).

**Output:** Fişierul de ieşire output.txt va conţine pe fiecare linie două numere separate prin spaţiu: numărul natural citit din fişierul de intrare şi echivalentul lui în baza 10.

**Exemplu:**:

|  |  |
| --- | --- |
| Fişierul **input.txt** | Fişierul **output.txt:** |
| 213  512  95  A8F | 213 39  512 188  95 95  A8F 2703 |

Problema 3. **Polinomul**

În fişierul *polinom.in* sunt scrişi coeficienţii a două polinoame, fiecare polinom este scris din linie nouă. Prima valoare din fiecare linie indică gradul polinomului (gradul polinomului 100). Următoarele valori indică coeficienţii polinomului aranjaţi în ordine descrescătoare conform gradului. De elaborat un program care va scrie în fişierul *polinom.out* suma pe prima linie şi diferenţa polinoamelor în linia doi, în formatul fişierului de intrare.

|  |  |
| --- | --- |
| *Exemplu puncte.in* | *puncte.out* |
| 3 2 1 -5 6  1 2 -3 | 3 2 1 -3 3  3 2 1 -7 9 |

Problema 4. P**etrolul**

Republica Moldova, trecută prin criza gazelor, şi neavînd resurse de petrol şi-a propus să procure petrol de peste hotare.

Există *n* ţări care exportă petrol. Fiecare *i*-stat exportă petrol cu *ai* dolari şi *bi*euro pentru un baril. Moldova nu poate procura petrol de la un stat şi în dolari şi în euro. Ajutaţi Republica Moldova să procure petrol cît mai mult, dacă se cunoaşte numărul *a*(numărul de dolari de care dispune RM) şi *b*(numărul de euro de care dispune RM).

**Input:** Fişierul de intrare *oil.in* conţine în primul rînd 3 numere: *n, a, b* (1 ≤ n ≤ 100, 0 ≤ a, b ≤ 1000). Următoarele *N* rînduri conţin perechile de numere *ai* şi *b*i

Output: Fiişierul de ieşire oil.out va conţine volumul maxim de petrol care va putea fi procurat cu minim 2 cifre după virgulă.

Exemplu:

|  |  |
| --- | --- |
| oil.in | oil.out |
| 3 2 5  6 4  3 5  8 7 | 1.92 |
| 4 3 2  1 1  2 2  3 3  4 4 | 4.00 |

Problema 5. **Evaluarea expresiilor**

Se consideră expresiile aritmetice formate din numere naturale şi operatorii binari + şi –. Scrieţi un program care evaluează expresiile aritmetice în studiu.

## Input: Fişierul INPUT.TXT va conţine expresia de evaluat.

## Output: Fişierul OUTPUT.TXT va conţine rezultatul evaluării.

## Restricţii: Numărul maxim de cifre al numerelor - 100.

## Exemplu: INPUT.TXT 93423432323232+233323439876343-34567886

### OUTPUT.TXT 326746837631689

Problema 6. **Cerc de suprafaţă maximă**

Pe o suprafaţă plană sunt date N puncte prin coordonatele lor (N999). De elaborat un program care va scrie în fişierul *puncte.out* aria cercului de suprafaţă maximă, interiorul căruia nu conţine nici un punct, cu excepţia centrului. În fişierul *puncte.in* este scris numărul N, după care, pe fiecare linie coordonatele fiecărui punct.

*Exemplu*

|  |  |
| --- | --- |
| *puncte.in* | *puncte.out* |
| 3  0 0  2 2  1.5 1.5 | 14,11 |
| Centru cercului de suprafaţă maximă este punctul (0;0), iar cel mai apropiat punct de către acesta este punctul (1,5;1,5). Prin urmare raza cercului este distanţa dintre punctele (0;0) şi (1,5;1,5). Determinând această distanţă aflăm şi aria cercului. | |

Pentru determinarea distanţei dintre 2 puncte utilizăm formula :



Problema 7. **Numere interesante**

Se ştie că de la numărul 4 se poate obţine orice număr natural scris în baza 10 prin aplicarea următoarelor operaţii:

1. Se scrie la sfârşit cifra 4;
2. Se adaugă la sfârşit cifra 0;
3. Se împarte la 2 (dacă numărul este par).

Să se scrie un program, care produce un şir de numere construit conform regulilor A – C, în care primul număr este 4 iar ultimul este N.

**Exemplu:** dacă N=15 calea este: 4 → C → 2 → A → 24 → C → 12 → C→ 6 → C → 3 → B→ 30 → C →15.

**Input:** Fişierul ***numere.txt*** va conţine pe unica linie - valoarea lui N

**Output:** Fişierul ***calea.txt*** va conţine calea obținută scrisă într-o linie fiind despărţite prin spaţiu. (Pentru exemplu 4 C 2 A 24 C 12 C 6 C 3 B 30 C 15).