

08-03-1.1

Pentru monitorizarea unor sere, un inginer folosește N senzori. Fiecare senzor are un ID de forma $SXXX$, unde XXX reprezintă numere (ex: $S001$, $S002$, $S023$ etc.). Fiecare senzor măsoară concentrația de CO_2 , temperatură și umiditate pentru o durată de M minute ($M < 60$).

Senzorii achiziționează date pe rând, de concentrație, temperatură și umiditate în fiecare minut.

Senzorii declanșează alarme dacă timp de 3 minute consecutive nu se respectă următoarele condiții:

- concentrație CO_2 între 6 și 10% ($6 \leq \text{conc. } CO_2 \leq 10$)
- temperatura între 15 și 25 ($15 \leq \text{temperatura} \leq 25$)
- umiditate ≤ 80 .

Sa se afișeze id-ul tuturor senzorilor care au declansat alarme. In cazul in care nu a fost declansat niciun senzor, se va afisa mesajul: „Functionare corecta”.

ID-ul senzorului, concentrația, temperatura și umiditatea sunt date private.

!!! Se vor utiliza concepte de clasă și obiect (constructor, funcții membre etc.)

!!! Nu se acordă punctaj pentru nerespectarea cerinței.

Date de intrare:

N, M

pe următoarele N linii: ID CO2_minut_0, Temp_minut_0,
Umiditate_temp_0, CO2_minut_1, Temp_minut_1 ... CO2_minut_M-
1, Temp_minut_M-1, Umiditate_temp_M-1

Date de iesire:

ID senzori cu alarme sau Functionare corecta

Exemplu:

Date de intrare:

3 7

S001 7 15 70 6 14 59 7 20 80 9 13 60 6 12 61 7 11 60 7 17 65

S002 7 16 79 8 16 80 8 17 76 7 18 75 8 19 76 9 18 80 8 19 81

S003 7 20 60 6 20 61 7 19 60 7 19 60 8 18 61 9 19 62 10 20 63

Date de iesire: S001

Explicatie: Temperaturile de la senzorul 1 (13, 12, 11) sunt in afara intervalului admis.

$N \leq 20$

$3 \leq M \leq 60$

08-03-1.2

Sa se scrie un program care contine o clasa, vect. Clasa vect contine datele membre: N (numarul de elemente din vector) si vectorul v, ce contine N valori reale.

Se vor defini constructorii si destructorii clasei.

Sa se implementeze urmatoarele functii pentru clasa vect:

void push(int i, double x) – functie care introduce valoarea x pe pozitia i a vectorului v

double pop(int i) – functie care permite extragerea valorii de pe pozitia i a vectorului v

double medie() – functie care returneaza media vectorului v

double stdev() – functie care returneaza deviatia standard a vectorului v

Programul va afisa la final media vectorului si deviatia standard a acestuia.

Nu se acordă punctaj pentru nerespectarea cerinței.

Date de intrare:

N

valorile vectorului

Date de iesire:

media deviatia standard

Exemplu:

Date de intrare:

5

1 3 4 6 8

Date de iesire:

4.4 2.41

Formula de calcul a deviatiei standard (u reprezinta media).

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}.$$

08-03-1.3

Turneul mondial de poker a început și vi se cere să dezvoltați un program care determină mâinile câștigătoare.

Un pachet standard folosit în poker conține 52 cărți. Fiecare carte are un simbol de treflă, romb, inimă sau pică (notate T, R, I, P în datele de intrare). Fiecare carte poate avea una din următoarele valori notate 2, 3, 4, 5, 6, 7, 8, 9, Z, J, Q, K, A, (Z reprezintă 10). Valorile cărților sunt ordonate după valoare, 2 având cea mai mică valoare și A(as) cea mai mare valoare. Simbolul nu are impact asupra valorii cărții dar poate avea impact asupra valorii mâinii.

O mână de poker este formată din cinci cărți distribuite din pachet.

Programul citește 5 cărți de intrare pentru fiecare mână sub formă de șiruri.

Converțiți șirul în numere și determinați tipul combinației de cărți.

Verificați care mână are o combinație mai bună. În caz de egalitate al combinației de cărți, câștigă combinația cu valoarea cea mai mare.

Ierarhia mâinilor în cazul programului este următoarea:

chintă de simbol (5 carti de valori consecutive si acelasi simbol)

careu (4 carti de aceeasi valoare)

full (3 carti de valoare identica si 2 cu valori identice diferite de primele 3)

simbol (5 carti cu acelasi simbol)

chintă (5 carti consecutive ca valoare dar de simboluri diferite)

trei (trei carti de aceeasi valoare)

pereche(2 carti de aceeasi valoare)

carte mare (atunci cand nu se formeaza niciuna din combinatiile de mai sus).

Determinati jucatorul cu mana cea mai buna.

!!! Se vor utiliza concepte de clasă și obiect (constructor, destructor, funcții membre etc.).

!!! Nu se acordă punctaj pentru nerespectarea cerinței.

Date de intrare:

Primele 5 valori (corespunzătoare primului jucător) urmate de alte 5 valori (corespunzătoare celui de-al doilea jucător)

Date de iesire:

1 dacă primul jucător câștigă

2 dacă al doilea câștigă

egal dacă este egalitate

Exemplul 1:

Date de intrare:

2T 2R 6P 7i KT 3i 3P 6T QT AR

Date de ieșire:

2

Explicație: Amândoi jucătorii au o pereche. Al doilea jucator are o pereche de 3 care este mai mare ca și valoare ca o pereche de 2.

Exemplul 2:

Date de intrare:

2T 3R 4P 5i 6R KT 9T 8T 7T 6T

Date de ieșire:

2

Explicație: Primul jucător are chintă (numere în ordine) dar al doilea jucător are simbol (5 carti cu acelasi simbol), deci are mâna mai valoroasă.

15-03-2.1

Fie clasa Polinom care implementeaza notiunea de polinom. Clasa are ca date private urmatoarele date: grad (gradul polinomului) si coef (coeficientii polinomului).

Determinati suma si produsul a doua polinoame.

Se vor utiliza urmatoarele supraincari: operator indexare, operatori aritmentici, operatori citire si afisare in stream.

Nerespectarea cerinței atrage anulara punctajului!

Date de intrare:

Gradul primului polinom
Coeficientii primului polinom

Gradul celui de-al doilea polinom
Coeficientii celui de-al doilea polinom

Date de iesire:

Suma
Produsul

Exemplu:

Date de intrare:

1

1 2

2

3 4 4

Date de iesire:

$4x^2+6x^1+4x^0$

$8x^3+12x^2+10x^1+3x^0$

15-03-2.2

Fie clasa `Numar_Mare` care implementează noțiunea de număr mare. Clasa are ca date private variabila cifre (reține cifrele numărului) și `nr_cifre` (reține câte cifre are numărul). Să se implementeze următoarele operații:

- determinați suma a două numere mari
- comparați cele două numere și îl afișați pe cel mai mare
- afișați primele 3 cifre ale celui mai mare număr

Se vor utiliza următoarele supraîncărcări: operator indexare, operator aritmetic, operator relațional, operatori de citire și afișare în stream.

Nerespectarea cerinței atrage anularea punctajului!

Date de intrare:

Numărul de cifre ale primului număr

Cifrele

Numărul de cifre ale celui de-al doilea număr

Cifrele

Date de ieșire:

Suma

Numărul cel mai mare

Primele cifre ale celui mai mare număr

Exemplu:

Date de intrare:

3

3 2 1

2

1 4

Date de ieşire:

335

321

3

2

1

30-03-3.1

Se consideră o clasă de bază Forma având ca date membre coordonatele a 4 puncte în plan x_A y_A x_B y_B x_C y_C x_D y_D . implementați clasele derivate Pătrat (derivată din Forma) și Dreptunghi (derivată din Forma). Adăugați funcțiile Arie(), Perimetru() și TipForma() pentru fiecare clasa derivată.

Se vor utiliza concepte de clasă și obiect.

Date de intrare:

Coordonatele punctelor

Date de iesire:

Aria

Perimetrul

Tipul formei

Exemplu:

Date de intrare:

1 1

4 1

4 3

1 3

Date de iesire:

6

10

Dreptunghi

30-03-3.2

Se definește o clasă Masina pentru caracterizarea unei mașini prin pret și autonomie. Se definește o a doua clasă Baterie definită prin capacitate și durată de viață. Pe baza acestor clase să se deriveze o clasă MasinaElectrica. În funcția main, după inițializarea unui obiect din clasa MasinaElectrica, se vor calcula folosind funcții membre:

- prețul după următoarea formulă
 $\text{preț} = \text{pret_masina} + \text{durata_viata} * 12,$
- autonomia după următoarea formulă
 $\text{autonomie} = (\text{capacitate} * \text{durata_viata}) / 100.$

Se vor utiliza concepte de clasă și obiect.

Nu se acroda punctaj pentru nerespectarea cerintei!

Date de intrare:

pret_masina autonomie
capacitate durata de viata

Date de iesire:

pret masina electrica
autonomie masina electrica

Exemplu:

Date de intrare:

100 1000

2200 3

Date de iesire:

136

66

30-03-3.3

Se definește o clasă Student cu date membre protejate nume, facultate, medie. Datele sunt protejate. Din clasa student se derivează clasa Absolvent. Datele membre pentru absolvent sunt: Licenta. Data este privata. Un student nu poate fi absolvent dacă nu are licenta, are media mai mică de 6 sau nu este înmatriculat la ETTI. Să se citească informațiile despre fiecare student și apoi să se afișeze dacă este absolvent.

Se vor utiliza concepte de clasă și obiect.

Nu se acordă punctaj pentru nerespectarea cerinței!

Date de intrare:

nume
facultate
medie

Date de ieșire:

DA - dacă este absolvent
NU - dacă nu este absolvent

30-03-1

Pentru rezolvarea unui sistem de trei ecuații liniare cu trei necunoscute prin metoda Cramer să se definească următoarele clase:

- clasa Vector, corespunzând vectorului coloană de dimensiune 3 a termenilor liberi dintr-un sistem, având constructori și funcții membre pentru citirea și afișarea componentelor;

- clasa Matrice, corespunzând matricii de dimensiune 3×3 a coeficienților necunoscutelor, având constructori și funcții membre pentru citirea componentelor, afișarea componentelor și calcularea determinantului corespunzător matricii;

- clasa Sistem, derivată din primele două, corespunzând sistemului complet, având constructori și funcții membre pentru înlocuirea unei coloane a matricii coeficienților necunoscutelor prin vectorul termenilor liberi și pentru rezolvarea sistemului.

Clasa Sistem va fi utilizată pentru rezolvarea unui sistem 3×3 prin metoda Cramer.

!!!Nu se acordă punctaj pentru nerespectarea cerinței.

Date de intrare:

matricea formată din coeficienți și termeni liberi

Date de ieșire:

soluțiile sistemului

Exemplu:

Date de intrare:

1 0 2 9
0 2 1 8
4 -3 0 -2

Date de ieşire:

1.00
2.00
4.00

30-03-2

Ati fost angajat ca si programator si aveti de dezvoltat jocul World of Warcraft.

Jucatorii au de ales intre cele 2 factiuni existente: Alianta (oameni) si Hoarda (orci). Jucatorii au de ales intre a lupta contra altor jucatori PVP sau a lupta in raid PVE. In functie de actiunea aleasa si attributele pe care le are, jucatorul este recompensat cu un numar de token, Valor Points. Pentru a participa in PVP jucatorii trebuie sa cheltuiasca un numar de Valor Points pentru a cumpara echipament mai bun si a-si imbunatati cu 1 punct un atribut.

Fiecare jucator are 3 attribute: HP (H), Mana (M), Level (L).

In PVE sunt disponibile urmatoarele optiuni:

- Icecrown Citadel (ICC) (disponibila pentru Alianta si Hoarda). Numarul de Valor Points care se poate obtine este egal cu $H+M+L$
- Naxxramas (NX) (disponibila doar pentru Hoarda). Numar de Valor Points care se obtine este $2M+L$
- Blackrock Depths (BK) (disponibila doar pentru Hoarda). Numar de Valor Points care se obtine $2H+M$
- Stratholme (SA) (disponibila pentru Alianta). Numar de Valor Points care se obtine $2H+L$
- Violet Hold (VH) (disponibil pentru Alianta). Numar de Valor Points care se obtine $2L+M$

In PVP echipamentele au costuri diferite in functie de atributul care trebuie imbunatatit astfel:

- HP Hoarda plateste $5+H/3$ Valor Points, Alianta plateste $8+H/4$ Valor Points
- Mana Hoarda plateste $3+M/7$, Alianta plateste $5+M/5$
- Level Hoarda plateste $5+L/6$, Alianta plateste $4+L/5$.

Suplimentar, jucatorii pot evolua astfel:

- un jucator din Alianta devine Grand Crusader daca participa de 4 ori in Violet Hold.
- un jucator din Hoarda devine Warlord daca participa in 2 ori in total in Naxxramas.

Initial jucatorii au 0 Valor Points si $H=M=L=1$.

Pentru Alianta: daca jucatorul devine Grand Crusader, se afiseaza 1, altfel 0.

Date de intrare:

numar jucatori Hoarda

numar jucatori Alianta

numar zile petrecute in joc

Pe urmatoarele linii (nr ore joaca*(jucatori Alianta + jucatori Hoarda))
avem actiunile desfasurate

Date de iesire:

informatii despre fiecare jucator (status update) dupa finalizarea unei actiuni

Exemplu:

Date de intrare:

2

1

3

ICC

NX

ICC

NX

M

VH

ICC

NX

H

Date de iesire:

1 1 1 3 0

1 1 1 3 1

1 1 1 3 0

1 1 1 6 1

1 2 1 0 1

1 1 1 6 0

1 1 1 9 1

1 2 1 5 2

1 1 1 6 0

Explicatie

prima zi

Prima linie - primul Orc merge in ICC si primeste 3 Valor Points.

Numarul de participari in Naxxramas este 0.

A doua linie - al doilea Orc merge in Naxxramas si primeste 3 Valor Points. Numarul de participari in NX este 1.

A treia linie - primul Om merge in ICC si primeste 3 Valor points.

Numarul de participar in VH este 0.

a doua zi

Linia 4 - primul Orc merge in NX si primeste 3 Valor points. Numarul de participari este 1.

Linia 5 - al doilea Orc imbunatateste Mana cu 1 punct. Noile attribute sunt 1 2 1 (H a ramas nemodificat adica 1, M a crescut cu 1 deci $1+1=2$,

L a ramas nemodificat deci 1). Foloseste punctele conform formulei si ramane cu 0. Se alege partea intreaga $[3+1/7]=3$. Numarul de participari in NX este neschimbat deci 1.

Linia 6 - primul Om participa in VH. Primeste Valor points si numarul de participari este 1. Cum nu se indeplineste conditia de Grand Crusader, se va afisa tot 0.

a treia zi

Linia 7 - primul Orc merge in ICC. Primeste 3 Valor Points. Numarul de participari in NX este in continuare 1.

Linia 8 - al doilea Orc merge in NX. Primeste 5 Valor Points. Numarul de participari in NX este 2.

Linia 9 - primul Om imbunatateste HP care costa 8 Valor Points.

Nu are destul Valor Point deci nu poate face aceasta actiune. Status ramane neschimbat.

Se vor utiliza concepte de clasa și obiect.

Se va utiliza mecanismul de mostenire.

Nu se acorda punctaj pentru nerespectarea cerintei!

12-04-4.1

Implementați o clasă Forma. Implementați două clase derivate din clasa de bază: Triunghi și Patrulater. Să se determine aria fiecarui poligon folosind pointeri la clasa de bază.

Se vor utiliza concepte de clasă și obiect.

Nu se acroda punctaj pentru nerespectarea cerintei!

Date de intrare:

baza

înălțimea

Date de ieșire:

arie patrulater

arie triunghi

Exemplu:

Date de intrare:

3

4

Date de ieșire:

12

6

12-04-4.2

Se considera o clasă de bază Conversie care convertește un număr din decimal în binar. Din clasa de bază se derivează două clase Hexa și Octal. Folosind funcții virtuale faceți conversia unui număr din decimal în hexazecimal și în octal. Apelarea și afisarea rezultatelor se vor face prin intermediul unui pointer redirecționat spre clasele derivate. Nu se acorda punctaj pentru nerespectarea cerinței!

numărul<256

Date de intrare:

numărul în decimal

Date de ieșire:

număr în binar

număr în hexa

număr în octal

Exemplu:

Date de intrare:

15

Date de ieșire:

00001111

F

17

12-04-4.3

Se consideră clasa Biblioteca cu N rafturi care permite adăugarea de cărți pornind de la raftul de sus în jos, și extragerea de cărți începând cu ultimul raft, de jos în sus. Să se deriveze 2 clase, Biblioteca_veche și Biblioteca_copii. Folosind conceptul de polimorfism să se implementeze două metode pentru adăugarea și extragerea cărților din clasele derivate astfel:

- În Biblioteca_veche cărțile se adaugă începând cu raftul de jos și se extrag începând cu ultimul raft ocupat
- În Biblioteca_copii cărțile se adaugă începând cu raftul de sus, în jos, iar extragerea se face începând cu primul raft de sus.

Cărțile sunt cunoscute după primul număr din ISBN ($N < 9999$)

La întâlnirea unui număr > 9999 se va efectua o extragere de carte.

Nu se acorda punctaj pentru nerespectarea cerinței!

Date de intrare:

Cărțile

Date de ieșire:

Cărțile din biblioteca veche

Cărțile din biblioteca copii

Exemplu:

Date de intrare:

1 2 3 4 10000 5

Date de ieşire:

5 3 2 1

2 3 4 5

10-05-5.1

Să se implementeze clasele Cerc și Dreptunghi derivate din clasa de bază FiguraGeometrica. Clasa FiguraGeometrica va fi definită ca și clasă abstractă având declarate metodele getPerimetru() și getAria() ca funcții virtuale pure.

Date de intrare:

Raza

Latura mica

Latura mare

Date de ieșire:

Perimetru cerc

Arie cerc

Perimetru dreptunghi

Arie dreptunghi

Exemplu:

Date de intrare:

1

2

3

Date de ieşire:

6.28

3.14

10

6

10-05-5.2

Se consideră clasa Biblioteca. Să se deriveze 2 clase, Literatura și Manuscrise. Folosind funcții virtuale pure să se implementeze două metode pentru adăugarea și extragerea cărților din clasele derivate astfel:

- În Literatura cărțile se adaugă începând cu raftul de jos și se extrag începând cu ultimul raft ocupat
- În Manuscrise cărțile se adaugă începând cu raftul de sus, în jos, iar extragerea se face începând cu primul raft de sus.

Cărțile sunt cunoscute după primul număr din ISBN ($N < 9999$)

La întâlnirea unui număr > 9999 se va efectua o extragere de carte.

Nu se acorda punctaj pentru nerespectarea cerinței!

Date de intrare:

Cărțile

Date de ieșire:

Cărțile din Literatura

Cărțile din Manuscrise

Exemplu:

Date de intrare:

1 2 3 4 10000 5

Date de ieşire:

5 3 2 1

2 3 4 5

10-05-5.3

Se consideră clasa Biblioteca. Să se deriveze 2 clase, Literatura și Manuscrise. Folosind conceptul de polimorfism să se implementeze două metode pentru adăugarea și extragerea cărților din clasele derivate astfel:

- În Literatura cărțile se adaugă începând cu raftul de jos și se extrag începând cu ultimul raft ocupat
- În Manuscrise cărțile se adaugă începând cu raftul de sus, în jos, iar extragerea se face începând cu primul raft de sus.

Să se implementeze o clasă Carti_Rare derivată din Literatura și Manuscrise. Adăugarea cărților în această secțiune se poate face fie adăugând de la raftul de sus în jos sau de la raftul de jos în sus. Extragerea se realizează după aceleași reguli ca și la adăugare.

Cărțile sunt cunoscute după primul număr din ISBN ($N < 9999$)

La întâlnirea unui număr > 9999 se va efectua o extragere de carte. Dacă numărul este par atunci extragerea din Carti_Rare se efectuează începând cu ultimul raft ocupat,. Dacă numărul este impar atunci extragerea din Carti_Rare se efectuează începând cu primul raft de sus.

La întâlnirea unui număr ≥ 100 adăugarea în Cărți_Rare se face începând cu raftul de sus, în jos altfel începând cu raftul de jos.

Nu se acroda punctaj pentru nerespectarea cerinței!

Date de intrare:

Cărțile

Date de ieșire:

Cărțile din Literatura

Cărțile din Manuscrise

Cărțile din Cărți_Rare

Exemplu:

Date de intrare:

1 2 3 4 10000 5

Date de ieșire:

5 3 2 1

2 3 4 5

5 3 2 1

10-05-5.4

Un student este definit prin Nume, CNP, punctaj la laborator, punctaj examen și rezultat_Engleza. Pentru a promova, studentul roman are nevoie de minim 50 puncte în total din laborator și examen, în timp ce studentul Erasmus are nevoie inclusiv de testul de la engleza. Dacă testul de la engleză este mai mic de 50 atunci indiferent dacă obține punctaj de promovare (≥ 50) va primi 0 puncte.

Studentii străini au prima cifră din CNP 7 sau 8.

Afișați studenții în ordinea descrescătoare a mediei. În caz de medii egale, departajarea se va face după rezultatul de la examen.

Date de intrare: Nume, CNP, punctaj laborator, punctaj examen, test engleza

Date de iesire: Studentii ordonati dupa medie

Exemplu:

Date de intrare:

Ion 1234567891011 10 45 0

Maria 2234567891011 45 37 15

Mike 7234567891011 25 25 60

Ela 8454567891011 45 35 45

Date de iesire:

Maria

Ion

Mike

Ela

Explicație: Maria are 82 puncte în total deci promovează. Ion are 55 puncte deci promovează. Ambii sunt români deci testul la engleză nu se ia în considerare.

Mike are 50 puncte la examen și 60 puncte la engleză. Având punctaj la engleză ≥ 50 , acesta promovează.

Ela are 45 puncte la engleză deci punctajul ei este 0.

10-05-1

O firmă de distribuție livrează în K țări. În fiecare zi, firma se află într-o țară și în ziua următoare poate ajunge în orice altă țară.

În fiecare țară există N resurse. Țările pot deține cantități din fiecare resursă și își pot modifica cantitatea de resurse deținute prin interacțiuni proprii sau prin interacțiune cu firma.

Interacțiunile proprii ale unei țări sunt:

- Producția: țara obține o cantitate X dintr-o anumită resursă;
- Consumul: țara consumă o cantitate dintr-o resursă. Dacă țara nu deține suficient din resursa respectivă atunci va consuma toată resursa respectivă și fericirea țării va scădea cu diferența;
- Prelucrarea : țara consumă o cantitate dintr-o resursă pentru a obține o cantitate din altă resursă. Dacă țara nu deține suficient din resursa consumată atunci prelucrarea nu va avea loc.

În ceea ce privește interacțiunile proprii, se cunosc următoarele următoarele:

- Nu este obligatoriu ca o țară să interacționeze cu toate resursele disponibile
- Nu va exista nici o resursă care să fie consumată în 2 interacțiuni diferite de către țara respectivă
- Nu va exista nici o resursă obținută din prelucrare care să fie consumată în altă prelucrare de către țara respectivă

Interacțiunile dintre o țară și firmă se realizează vânzare / cumpărare.

Țara sau firma oferă bani pentru o cantitate dintr-o resursă deținută de cealaltă parte.

Ofertele la care are acces țara în momentul când firma este prezentă sunt:

- Cumpără: firma oferă o cantitate dintr-o resursă. Firma deține resursa în cantitatea pe care o oferă. Țara poate să accepte oferta sau nu.
- Vinde: firma cere o cantitate dintr-o resursă. Țara poate să dețină resursa în cantitatea cerută sau nu. Țara poate să accepte oferta sau nu.

Dacă o ofertă a fost acceptată, cantitatea efectivă ce va fi schimbată între cele 2 părți implicate depinde de banii celui care plătește (dacă nu poate plăti toată cantitatea, va primi o cantitate întreagă egală cu cât poate plăti).

În ceea ce privește interacțiunile țării cu firma, acestea sunt:

- Cumpără: doar resursele de consum / doar resursele de producție + consum / toate resursele
- Vinde: orice resurse / orice resurse mai puțin cele de consum / orice resurse mai puțin cele de producție + consum

Toate ofertele se realizează în ordinea în care au fost primite.

Prețul de bază al resurselor este același pentru toate țările. Prețul efectiv al unei resurse depinde de cantitatea de resursă deținută de țară astfel:

- Preț cumpărare = preț bază - 3 - [cantitate / 5] ([] partea întreagă).
- Preț vânzare = preț bază - [cantitate / 5] ([] partea întreagă).

Prețul efectiv va fi întotdeauna cel puțin egal cu 1 și va fi constant pe durata unei zile.

Fericirea țării:

- este număr întreg.
- scade pentru fiecare interacțiune de consum incompletă cu cantitatea lipsă (ex: țara vrea să consume 5 PESTE, dar deține doar 3 PESTE, deci pierde 2 fericire).
- crește cu 5 la sfârșitul fiecărei zile.
- la sfârșitul zilei, (după creșterea cu 5 puncte) dacă fericirea este mai mare strict decât 100, ea devine 100
- la sfârșitul zilei, (după creșterea cu 5 puncte) dacă fericirea este mai mică strict decât 50, ea devine 50
- fericirea (F) influențează proporțional cantitatea (C) de resurse / bani obținuți zilnic după formula $[C * F / 100]$ ([] partea întreagă).

În fiecare zi, pentru fiecare țară, se realizează următoarele acțiuni în această ordine:

1. Țara primește bani din taxe.
2. Dacă firma se află în țară, se rezolvă ofertele de cumpărare.
3. Se realizează toate producțiile.
4. Se realizează toate prelucrările.
5. Se realizează tot consumul.
6. Dacă firma se află în țară, se rezolvă ofertele de vânzare.

Înainte de prima zi firma deține B bani și nici o resursă.

Înainte de prima zi, fiecare țară deține 0 bani și nici o resursă. De asemenea, fiecare țară are fericirea 100.

Simulați primele D zile și afișați la sfârșitul fiecărei zile informațiile despre țări.

Date de intrare

N numărul de resurse

numele celor N resurse

prețul celor N

B numărul inițial de bani ai firmei.

K numărul de țări.

cele K țări sub forma:

Pe prima linie NUME_ȚARA BANI_TAXE C_CUMPARARE ***
C_VANZARE ***, reprezentând numele țării, banii obținuți zilnic din
taxe, comportamentul la cumpărare și comportamentul la vânzare.

*** C_CUMPARARE este unul dintre {CONSUM, PRODUCTIE,
TOT} (cumpără doar resurse pentru consum, doar resurse pentru consum
+ prelucrare sau cumpără tot ce i se oferă).

*** C_VANZARE este unul dintre {ORICE, FARA_CONSUM,
FARA_PRODUCTIE} (vinde orice, orice mai puțin resursele de
consum, orice mai puțin resursele de consum + prelucrare).

Pe următoarea linie N_PRODUCTIE, numărul de interacțiuni de
producție.

Urmează N_PRODUCȚIE linii ce descriu interacțiunile de producție pentru țară sub forma:

RESURSA_OBTINUTA CANTITATE_OBTINUTA

Pe următoarea linie N_PRELUCRARE, numărul de interacțiuni de prelucrare.

Urmează N_PRELUCRARE linii ce descriu interacțiunile de prelucrare pentru țară sub forma:

RESURSA_CONSUMATA CANTITATE_CONSUMATA
RESURSA_OBTINUTA

CANTITATE_OBTINUTA

Pe următoarea linie N_CONSUM , numărul de interacțiuni de consum.

Urmează N_CONSUM linii ce descriu interacțiunile de consum pentru țară sub forma:

RESURSA_CONSUMATA CANTITATE_CONSUMATA

Pe următoarea linie D, numărul de zile ce trebuie simulate.

Se descriu în continuare cele D zile, sub forma:

Pe prima linie: NUME_TARA, țara în care se afla firma în ziua respectivă.

Pe următoarea N_CUMPARARE numărul de oferte de cumpărare ale firmei.

Urmează N_CUMPARARE linii ce descriu ofertele de cumpărare sub forma RESURSA CANTITATE

Pe următoarea N_VANZARE numărul de oferte de vanzare ale firmei.

Urmează N_VANZARE linii ce descriu ofertele de vânzare sub forma
RESURSA CANTITATE

Date de ieșire

La finalul fiecărei zile, informații despre toate țările, în ordinea în care
au fost date acestea, astfel:

pe prima linie, numele țării, fericirea și numărul de bani pe care îi deține
țara.

Pe următoarele linii, în ordine alfabetică a resurselor, numele resursei și
cantitatea deținută doar dacă țara deține resursa respectivă (are cel puțin
1 cantitate din ea). Afișați un rând liber după fiecare țară. Forma afișării:

NUME_ȚARA FERICIRE_ȚARA BANI_ȚARA

RESURSA_DETINUTA_1 CANTITATE_RESURSA_1

...

RESURSA_DETINUTA_K CANTITATE_RESURSA_K

Exemplu:

Date de intrare:

5

LEMN PIATRA ARGILA FIER MANCARE

10 10 10 10 10

100

2

ROMANIA 30 CONSUM FARA_CONSUM

2

MANCARE 11

PIATRA 3

1

PIATRA 5 FIER 2

2

MANCARE 10

LEMN 9

BULGARIA 10 PRODUCTIE ORICE

2

LEMN 10

ARGILA 10

1

PIATRA 5 FIER 2

1

MANCARE 9

5

ROMANIA

0

0

ROMANIA

0

0

ROMANIA

0

0

ROMANIA

0

0

ROMANIA

0

0

Date de iesire:

ROMANIA 96 30

MANCARE 1

PIATRA 3

BULGARIA 96 10

ARGILA 10

LEMN 10

ROMANIA 92 58

FIER 1

MANCARE 1

BULGARIA 92 19

ARGILA 19

LEMN 19

ROMANIA 88 85

FIER 1

MANCARE 1

PIATRA 2

BULGARIA 88 28

ARGILA 28

LEMN 28

ROMANIA 84 111

FIER 1

PIATRA 4

BULGARIA 84 36

ARGILA 36

LEMN 36

ROMANIA 79 136

FIER 2

PIATRA 1

BULGARIA 80 44

ARGILA 44

LEMN 44

10-05-2

Se consideră o clasă abstractă Semnal care permite adăugarea eşantioanelor în mod succesiv. Să se deriveze clasele DSP1 şi DSP2 din clasa de bază.

Să se implementeze metode de prelucrare în funcţie de tipul semnalelor astfel:

- în DSP1 se va extrage numărul de treceri prin zero ale semnalului, iar în DSP se vor extrage numărul de maxime ale semnalului
- DSP1 va modifica semnalul astfel încât esantionul curent să reprezinte media ultimelor 9 esantioane, iar DSP2 va determina câte maxime sunt mai mari decât tensiunea de prag (tensiunea de prag P va fi dată de la tastatură)
- DSP1 va pragui semnalul (tensiunea de prag este egală cu 15) astfel încât orice esantion ≥ 15 devine 1 şi orice esantion < 15 devine 0, iar în DSP2 maximele care sunt sub tensiunea de prag sunt înlocuite cu media aritmetică a valorilor vecine (valoarea anterioară şi valoarea următoare)
- DSP1 va subesantioneze semnalul cu factor de 5 (este păstrat fiecare esantion din 5 în 5 începând cu primul) iar DSP2 va insera la începutul şi la finalul semnalului câte 5 esantioane de gardă cu valoarea 0.

Nu se admit rezolvări bazate pe vectori.

Date de intrare:

tipul semnalului (1 sau 2)

tensiunea de prag P (se va lua în considerare doar dacă este semnal de tipul 2)

esantioanele semnalului

Date de iesire:

semnalele rezultate in urma fiecarei operatii de prelucrare

Exemplu:

Date de intrare:

1

10

-1 3 15 8 2 -7 -10 -3 6 21

Date de ieşire:

3

-1 3 15 8 2 -7 -10 -3 6 1.44 (primele 9 esantioane nu pot fi calculate ca si medie deci doar esantionul 10 poate fi calculat)

0 0 1 0 0 0 0 0 0 0

-1 -7 (subesantionam cu factor 5 incepand cu primul esantion)

24-05-1 & 24-05-2

Să se implementeze un program care administrează un card de călătorii pentru transport public.

Tranzacțiile posibile sunt:

A – adaugă. Suma de bani care se adaugă pe card.

M – călătorie metrou. Valoarea unei călătorii cu metroul este 3 lei. Valoarea călătoriilor efectuate se scade din suma de pe card. Nu se pot efectua călătorii cu metroul dacă suma de pe card este mai mică decât valoarea unei călătorii (3 lei).

S – călătorie STB. Valoarea unei călătorii este 1.5 lei. Valoarea călătoriilor efectuate se scade din suma de pe card. Se pot efectua călătorii cu STB dacă suma de pe card este mai mică decât valoarea unei călătorii (1.5 lei) dar se achită o amendă de 100 lei în ziua respectivă.

T – călătorie taxi. Tariful de călătorie este de 2 lei/km. Valoarea călătoriilor efectuate se scade din suma de pe card și se percepe un comision de 2.5% din valoarea totală a călătoriei. Nu se pot efectua călătorii cu taxi dacă suma de pe card este mai mică de 10 lei sau valoarea cardului devine negativă.

Cardul este creat în ziua 0 și are 0 lei.

Rezolvarea trebuie să conțină noțiuni de POO: clase derivate, metode virtuale, polimorfism.

Sugestie de implementare (nu este obligatorie respectarea sugestiei date):

Clasă de bază Card din care se derivează clasele pentru operațiuni.

Subiect 1) Determinați operațiunea care modifică cel mai mult valoarea cardului în ziua în care este efectuată operațiunea. – **24-05-1**

Subiectul 2) Valoarea cadrului în ziua Z – **24-05-2**

Date de intrare:

N- număr de tranzacții

Z – ziua

Pe următoarele N linii

A suma ziua

M număr_călătorii ziua

S număr_călătorii ziua

T km_parcurși ziua

Date de ieșire:

Subiectul 1) – Numărul de ordine al tranzacției

Subiectul 2) – Valoarea cardului afișată cu 2 zecimale

28-05-1 & 28-05-2

Folosind principiile POO (derivare, moștenire, virtualizare, polimorfism) să se rezolve următoarea problemă:

Să se implementeze un sistem de plată comun pentru transportul persoanelor cu companiile RATB sau Metrou.

Fiecare călător are un ID asociat, valoare întreaga pe 64 biți fără semn. Persoanele pot cumpăra abonamente sau pot plăti cursele individual. Suntem interesați să știm câți bani cheltuie fiecare călător.

Atât la RATB cât și la Metrou abonamentele și biletele au aceleași categorii posibile:

- (a) Abonament pentru o luna, având prețul P_{a1L}
- (b) Abonament pentru 3 zile cu prețul P_{a3Z}
- (c) Abonament pentru 24 ore cu prețul P_{a24O}
- (d) Bilet de călătorie valabil 90 minute = P_{b90min}
- (e) Bilet de călătorie pentru 1 călătorie = P_{b1C}
- (f) Bilet de călătorie pentru un număr N de stații = $P_{bNS} * N$

Prețurile de la RATB și de la Metrou sunt diferite pentru fiecare categorie și se citesc de la tastatură. Pentru împărțirea veniturilor între cele 2 companii se va ține cont de compania cu care se face călătoria.

În momentul în care o persoană dorește să facă o călătorie se verifică mai întâi dacă are un abonament valabil. Dacă nu are abonament, se cumpără biletul specificat. Un călător NU trebuie să mai cumpere un bilet valabil 90 de minute dacă are deja unul valabil la compania respectivă.

Pentru simplificare, zilele se vor număra continuu, începând cu 1 ianuarie până la sfârșitul anului (valori întregi de la 1 la 365). Ora (timpul din zi) se va număra continuu în minute (de la 1 până la 1440 inclusiv).

Date de intrare:

Pe prima linie, lista de prețuri pentru RATB în ordinea:

P_a1L-RATB P_a3Z-RATB P_a24O-RATB P_b90min-RATB P_b1C-RATB P_bNS-RATB

Pe a doua linie, lista de prețuri pentru Metrou în ordinea:

P_a1L-M P_a3Z-M P_a24O-M P_b90min-M P_b1C-M P_bNS-M

Urmează un număr necunoscut de linii ce descriu acțiunile călătorilor. Acestea pot fi de forma:

ZI ORA ID TIP_ABONAMENT COMPANIE

ZI ORA ID TIP_CALATORIE COMPANIE

Prima linie reprezintă acțiunile în care călătorul cu ID cumpără un abonament de tip TIP_ABONAMENT la COMPANIE.

TIP_ABONAMENT poate fi {A1L, A3Z, A24O} iar COMPANIE poate fi {RATB, M}.

A doua linie reprezintă acțiunile în care călătorul cu ID începe o călătorie de tip TIP_CALATORIE la COMPANIE. TIP_CALATORIE poate fi {C1, C90, C N}. În ultimul caz N este un număr natural.

Observații:

Toate prețurile sunt numere reale. Folosiți tipul de date float pentru a le reprezenta. Un călător nu plătește nici un fel de călătorie, dacă la momentul călătoriei deține un abonament valabil la acea companie.

Un călător nu plătește o călătorie de 90 de minute dacă deja deține un bilet de 90 de minute valabil la acea companie, dar va plăti normal celelalte călătorii la acea companie.

Un abonament lunar cumpărat de exemplu în ziua 10 la ora 900 va fi valabil până în ziua 40 ora 899 inclusiv.

Un abonament de 3 zile cumpărat de exemplu în ziua 5 la ora 1200 va fi valabil până în ziua 8 ora 1199 inclusiv.

Un abonament de 24 ore cumpărat de exemplu în ziua 12 la ora 400 va fi valabil până în ziua 13 la ora 399 inclusiv.

Un bilet de 90 de minute cumpărat de exemplu în ziua 13 la ora 60 va fi valabil pentru călătorii de 90 de minute până în ziua 13 la ora 149 inclusiv.

În datele de intrare, data (ZI, ORA) se garantează ca va fi crescătoare.

Între 2 folosiri ale aceluiași bilet de 90 de minute pentru un călător pot exista și alte călătorii care să nu fie de 90 de minute, inclusiv la firma cealaltă.

Cerințe:

(a) Să se afișeze, separate prin spațiu, id-ul și cheltuielile pentru persoana care a cheltuit cel mai mult pe transport în întreaga perioadă a datelor de intrare. Cheltuielile se vor afișa cu exact 2 zecimale.:

ID CHELTUIELI

(b) Să se afișeze, separate prin spațiu și cu 2 zecimale, suma câștigată de compania RATB și suma câștigată de compania de Metrou:

CASTIG_RATB CASTIG_M

Se vor lua în considerare NUMAI implementările care folosesc principiile POO. Celelalte soluții se vor anula.

02-06-1

Folosind **derivare, moșteniri, virtualizare și polimorfism** să se implementeze un program care gestionează serviciile oferite de companie de telefonie mobilă.

Serviciile oferite sunt:

A – Abonament Standard. Costul abonamentului este de 50 lei. Abonamentul include 100 minute, 100 SMS și 10GB internet. Dacă sunt depășite cele 100 minute utilizatorul plătește 0.1 lei pe minut în plus consumat. Dacă sunt depășite mesajele, clientul plătește 0.5 lei pe SMS în plus. Dacă este depășită limita de internet, utilizatorul plătește 1 leu pentru fiecare GB în plus.

B – Abonament Standard Plus. Costul abonamentului este de 75 lei. Abonamentul include 200 minute, 200 SMS și 15GB. Dacă sunt depășite cele 200 minute utilizatorul plătește 0.15 lei pe minut în plus consumat. Dacă sunt depășite mesajele, clientul plătește 0.75 lei pe SMS în plus. Dacă este depășită limita de internet, utilizatorul plătește 1.5 leu pentru fiecare GB în plus.

Clienții își achiziționează un abonament pe baza ID-ului propriu care este un număr pe 64 biți.

Din N înregistrări din gestiune, determinați ID-ul clientului care plătește cel mai mult pentru serviciile oferite. Un client poate avea unul sau mai multe abonamente de același fel sau diferite.

Sugestie de implementare:

Clasă de bază Servicii din care se derivează clasele pentru abonamente.

Date de intrare:

N – numărul de înregistrări pentru care se face analiza

ID_client Tip_Abonament Minute_folosite_total SMS_folosite_total
GB_folosit total

Date de ieșire:

ID-ul clientului care plătește cel mai mult

Exemplu:

Intrare:

3

10 A 100 100 10

20 B 200 200 15

30 A 100 150 15

Ieșire:

30

Explicație:

Clientul cu ID 10 folosește doar ce este inclus deja în abonament deci plătește doar prețul abonamentului 50 lei

Clientul cu ID 20 folosește doar ce este inclus deja în abonament deci plătește doar prețul abonamentului de tip B 75 lei

Clientul cu ID 30 depășește ce este inclus în abonamentul de tip A astfel că acesta are de plătit

$50 + (150-100)*0.5 + (15-10)*1 = 80$ lei