Pentru fiecare dintre cei n elevi ai unei clase se cunoaşte numărul matricol (format din maximum 8 cifre), anul, luna şi ziua naşterii. Să se afişeze lista elevilor care îşi vor sărbători ziua de naştere anul acesta, cunoscând ziua şi luna în care ne aflăm. Se vor afişa numărul matricol al elevului, ziua şi luna naşterii acestuia, în ordine crescătoare a datei naşterii. Datele se ctesc din fişierul *date.in* în care, pe prima linie se află n, ziua curentă şi luna curentă, iar pe următoarele n linii datele elevilor: numărul matricol, ziua, luna şi anul naşterii.
Exemplu: *date.in* se va afişa pe ecran
4 12 3 12301 21 4
23128 4 12 1998 23128 4 12
12301 21 4 1999
54312 11 3 1998
21398 23 1 1999

Variabila **x** este declarată alăturat. Indicați o instrucțiune de atribuire corectă din punctul de vedere sintactic.

```
struct masina
{ char serie;
int rating;
}x;
```

a. `x.rating=x.serie-'A';` **b.** `x.masina.rating=2021;`
c. `x(serie)='A';` **d.** `x=(2021,'A',10);`

Variabila **y** memorează simultan datele fiecăruia dintre cei **30** de elevi dintr-o clasă: codul de identificare la un examen (un număr natural din intervalul **[1,103))** și două note obținute (numere reale). Expresiile C/C++ de mai jos au ca valori codul de identificare și cele două note ale celui de al treilea elev din clasă. Scrieți definiția unei structuri cu eticheta **elev**, care permite memorarea datelor despre un elev, și declarați corespunzător variabila **y**.

y[2].cod **y[2].nota1** **y[2].nota2**

Fiecare dintre variabilele declarate alăturat memorează numele şi nota câte unui elev. Scrieţi secvenţa de instrucţiuni prin care se citesc de la tastatură numele şi nota pentru fiecare dintre variabilele **e1** şi **e2** şi apoi se afişează numele elevului cu nota cea mai mare.
Dacă cele două note sunt egale, se va afişa numele elevului memorat în variabila **e1.**

```
struct elev
{
char nume[20];
float nota;
};
elev e1,e2;
```

Se consideră tipul de date **punct** definit astfel

```
struct punct
{
   float x, y;
};
```

unde **x** și **y** reprezintă coordonatele carteziene ale unui punct**.**
Subprogramul **distanta** are doi parametri **A** și **B** de tip **punct** prin care primește coordonatele punctelor **A** și **B** în plan.
Subprogramul returnează distanța dintre punctele **A** și **B**.
Scrieţi la calculator definiţia completă a subprogramului.

a) Fișierul **puncte.in** conține pe prima linie numărul real nenul **r** și numărul natural **n (2<n<100**). Pe fiecare dintre următoarele **n** linii este scrisă câte o pereche de numere reale **x** și **y**, reprezentând coordonatele carteziene ale unui punct în planul **xOy**.

Se cere scrierea și executarea la calculator a unui program care citește din fișierul **puncte.in** numerele **r**, **n**, apoi cele **n** perechi de numere reale și afișează pe ecran coordonatele punctelor aflate pe cercul cu centrul în originea sistemului de coordonate și raza **r** sau mesajul **nu exista** în cazul în care niciunul dintre puncte nu îndeplinește această condiție. Valorile se afișează pe rânduri diferite, în format **(abscisă, ordonată)**. Pentru afișarea valorilor cerute se folosesc apeluri utile ale subprogramului **distanta.**

Exemplu: dacă fișierul **puncte.in** conține numerele

```
5
6
0 2
5 0
3 2
4 3
5 3
0 -5
```

atunci se afișează:

```
(5,0)
(4,3)
(0,-5)
```