

1. FP (circa 20')

Scrieți o funcție care satisface următoarea specificație:

Date $n, L=(l_1, l_2, \dots, l_n)$;

Precondiția: $l_i \in \mathbb{R}, i=1, n$

Rezultate $R=(r_1, r_2, \dots, r_n)$;

Postcondiția: R este o permutare a lui L , $r_1 \geq r_2 \geq \dots \geq r_n$.

2. OOP (circa 15')

Pentru gestiunea evenimentelor personale, implementați următoarele într-unul dintre limbajele C++, Java sau C#.

- O clasă *Event* ce reprezintă un eveniment personal având un singur atribut, numele evenimentului (*name*) reprezentat ca șir de caractere, și o singură metodă, *print* ce afișează evenimentul (numele lui) pe ieșirea standard.
- O clasă *Appointment* derivată din *Event* având un singur atribut, *startDate* reprezentat de asemenea ca șir de caractere, și o metodă *print* ce suprascrie metoda cu același nume din clasa de bază.
- Un program ce creează evenimentul "Absolvire facultate" și întâlnirea ("Licența, examen scris", "Luni, de la 8"), după care tipărește obiectele create.

2. SD (circa 12')

Identificați tipul abstract de date (STIVA, COADA, LISTA) potrivit pentru rezolvarea următoarei probleme și scrieți un algoritm pentru rezolvarea acesteia folosind operațiile din interfața tipului abstract de date (făcând abstracție de implementarea concretă a operațiilor). Se dă un șir de paranteze. Se cere să se verifice dacă parantezele se închid corect. De exemplu: $((()))$ se închid corect, $()()()$ nu se închid corect. Se vor specifica operațiile din interfața tipului abstract de date (fără implementarea acestora), se vor specifica și implementa subalgoritmii utilizați.

4. BD (circa 12')

- A. Se cere o bază de date relațională, cu tabele în 3NF, ce gestionează următoarele informații dintr-o facultate:
- discipline: cod, denumire, număr credite;
 - studenți: cod, nume, data nașterii, grupa, anul de studiu, specializarea, lista disciplinelor la care a dat examene (inclusiv data examenului și nota obținută).
- B. Pentru baza de date de la punctul 1, folosind algebra relațională **sau** instrucțiunile SELECT-SQL, se cer:
- Disciplinele pentru care nu există note de promovare (o disciplină este promovată dacă are cel puțin o notă ≥ 5).
 - Grupele care au cel puțin 20 de studenți.

5. SO (circa 12')

- A. Prezentați pe scurt stările READY și RUN ale unui proces și evenimentele care cauzează trecerea unui proces dintr-una într-alta.
- B. Explicați funcționarea scriptului sh de mai jos și semnificația rezultatului tipărit pe ecran. Modificați scriptul astfel încât rezultatul tipărit pe ecran să se aplice doar fișierelor cu mai mult de 10 linii.

```
1.      N=0
2.      for F in *.txt; do
3.      K=`wc -l $F|cut -d" " -f1`
4.      N=`expr $N + $K`
5.      done
6.      echo $N
```

Barem 1. FP:

- 1p oficiu
- 2p alegerea corectă a parametrilor de intrare, ieșire
- 5p implementarea unui algoritm de sortare în pseudocod sau într-un limbaj de programare
- la alegere)
- 2p stil: corectitudine, indentare, denumiri variabile, comentarii în cod.

Barem 2. OOP:

- Oficiu - 1p
- Clasa Event - 3p
- Clasa Appointment - 4p
- Program - 2p

Barem 3. SD:

- Oficiu 1p
- Identificarea Tipului Abstract de Date 1p
- Specificarea operațiilor folosite din TAD 2p
- Specificarea subalgoritmilor din algoritm 1p
- Scrierea subalgoritmilor folosiți în algoritm
- (implementarea subprogramelor folosite în aplicație) 3p
- Algoritmul/programul principal 1p
- Stil 1p

Barem 4. BD:

- 1 punct din oficiu
- Problema A: 2 puncte pentru tabele în 3NF; 1 punct pentru justificare
- Problema B: 3 puncte pentru A.i; 3 puncte pentru B.ii

Barem 5. SO:

- Oficiu 1p
- A 3p
- B Explicare si functionare 3p
- C Modificare (conditionarea executiei cut cu un if la linia 3) 3p