1. Ce se afiseaza la rularea codului C de mai jos ?
Cate procese au aparut in total (incluzand si procesul initial) ?
Desenati schitat arborescenta acestor procese.
Justificati raspunsurile.

```
int x=0;
for(fork(); fork(); exit(x))
 for(fork(); exit(x); fork())
  for(exit(x); fork(); fork())
          printf("%d\n", ++x);
```

2. Implementati in pseudocod o bariera de N procese folosind semafoare.
Detalii tehnice:
- se presupun predefinite:
constanta simbolica intreaga N;
tipul semafor "semaphore"
apelurile "up(&s)", "down(&s)", implementand operatiile uzuale de tip "up" si "down" asupra unui semafor "s";
se plot declara semafoare individuale sau vectori de semafoare, cu sau fara initializare; ex "semaphore s1, s2 s3[N], s4[2] = {0, 0};" samafoarele (individuale sau vectori) declarate fara initializare se presupun initializate cu 0;
- raspunsul se va redacta in limbajul C sub forma:

```
/* Date partajate */
 ..... (declaratii de semafoare, variabile, etc.)

/* Functia bariera executata in pro
```

3. Un sistem de programare in timp real are patru evenimente periodice cu perioadele de 50, 100, 200, si respectiv 250msec. Presupunem ca cele patru evenimente au nevoie de 35, 20, 10, si respectiv x msec timp de procesor. De asemenea, presupunem ca supraincarcarea cauzata de comutarea intre procese este neglijabila. Care este valoarea maxima a lui x pentru care sistemul este planificabil ?
Justificati raspunsul (aratati calculul).

4. Scrieti o functie in limbajul C:

```
 int upcase(char *fis);
```

care inlocuieste in fisierul 'fis' fisecare litera mica cu litera mare corespunzatoare (restul caracterelor raman nemodificate); functia returneazanumarul inlocuirilor facute sau -1 in caz de eroare.
Se va opera direct pe fisierul respectiv (in particular, nu se vor folosi vectori sau fisiere auxiliare). Se vor indica fisierle header necesare.
Indicatie: literele mici au coduri ASCII consecutive, la fel si literele mari, iar diferenta dintre codul ASCII al unei litere mici si cel al literei mai corespunzatoare este 32 (codul blankului); ex 'a'-'A'==' '.

5. Scrieti o functie in limbajul C:

int signfile(char *director);

care, pentru fiecare fisier obisnuit (regular) din directorul 'director', ce are anul ultimei actualizari egal cu anul curent din sistem, scrie la sfarsitul fisierului numele proprietarului acestuia. Functia returneaza numarul fisierelor semnate in acest fel, sau -1 in caz de eroare. Se vor indica fisierele header necesare.

6. Scrieti un program care efectueaza urmatoarele:
- primeste ca argument in linia de comanda in intreg n; daca nu nu este in intervalul 1, ..., 10, programul se termina cu eroare;
- aloca un tub anonim si genereaza n procese copil conectate la tub a.i. parintele sa citeasca din tub prin standard input iar copiii sa scrie in el prin standard output; procesele isi vor inchide ceilalti descriptori la tub;
- parinitele trimite fiecarui copil cate un semnal, ales aleator dintre semnalele 2,3, apoi asteapta terminarea tuturor copiilor.
- la primirea unui semnal, copilul scrie valoarea lui in tub, apoi se termina;
- dupa terminarea tuturor copiilor, parintele citeste cei n intregi din tub si-i afiseaza pe standard output