

## Indicații

- Testul conține 2 subiecte și durează 60 de minute. Fiecare subiect se punctează cu maxim 10 puncte pentru o rezolvare corectă și completă.
- Se pot obține punctaje parțiale pentru un subiect numai dacă acest lucru este specificat.
- Pentru a fi punctată, o rezolvare **trebuie** să includă și metoda de verificare a funcționalității acesteia.
- Pentru subiectul 2, este recomandată utilizarea macroului **PRINTF32** pentru afișarea la ecran, **doar dacă nu este explicit impusă utilizarea funcției printf()** de către enunț.
- Ordinea de rezolvare a subiectelor este la alegerea voastră.
- Subiectele se rezolvă pe mașina virtuală de PCLP2.

## Subiecte

### Subiectul 1

*With power comes great responsibility* spune vechea expresie englezescă. A venit vremea să vă responsabilizați în folosirea limbajului vostru preferat, C.

- [2p] Dându-se o variabilă de tip **signed short** cu valoarea inițială **-123**, adunați-o cu o altă valoare, pentru ca prin underflow, să ajungă la valoarea finală **123**.
- [3p] Fiind dat un pointer p, ce a fost referențiat de multiple ori, dereferențiați-l până ajunge să conțină string-ul **STOP**. Afișați numărul total al dereferențierilor.
- [2p] Folosind GDB afișați valoarea conținută de câmpul **age** al simbolului global **s**. **s** este de tipul **student\_t**:

```
struct student_t {  
    int id;  
    char name[20];  
    short age;  
};
```

- [3p] Pentru ierarhia conținută în directorul **my-hierarchy**, determinați numele tuturor simbolurilor neexportate aflate în zona **.bss**.

### Subiectul 2

Argumentele în linie de comandă dau bătăi de cap, dar sunt o modalitate consacrată de a trimite input diferitelor programe. Este timpul să le dați și voi de cap, în assembly, bineînțeles!

- [3p] Creați funcția **check\_controlsum** cu semnătura **int check\_controlsum(char \*s, int len)**, care verifică dacă un string are proprietatea că toate caracterele aflate pe poziție oglindită față de mijlocul stringului prin adunare rezultă în overflow (pe tipul de date **signed char**). În cazul în care proprietatea se verifică funcția întoarce valoarea 1, altfel aceasta întoarce valoarea 0.

Spre exemplu, pentru sirul

- **a byz**, funcția întoarce valoarea 1 (**a + z** și **b + y** rezultă în overflow).
  - **y12a**, funcția întoarce valoarea 0 (**y + a** rezultă în overflow, dar **1 + 2** nu rezultă în overflow).
- [4p] Salvați local în funcția **main** stringurile “Ana”, “are”, “alune”, “Buun3” și folosiți funcția definită la sub-punctul a) pentru a le verifica că au proprietatea de overflow. Dacă toate cuvintele au această proprietate, afișați mesajul *Sentence checks*, altfel mesajul *Sentence doesn't check*.
  - [3p] Folosiți funcția definită la subpunctul a) pentru a verifica dacă al doilea argument în linia de comandă al programului curent are proprietatea de overflow. Dacă programul nu este chemat cu cel puțin 2 argumente, printați mesajul *Not enough arguments*, altfel printați mesajul *My check is:* urmat de rezultatul întors de funcția **check\_controlsum**.

Spre exemplu, pentru invocarea programului

- **./file Maria**, se printează mesajul *My check is: 1*.
- **./file**, se printează mesajul *Not enough arguments*.