

## Test practic APD

Să se implementeze, **folosind Pthreads**, un algoritm paralel pentru determinarea numărului de apariții ale unei valori într-un vector cu N elemente, precum și a sumei pozițiilor la care valoarea se găsește în vector, folosind P thread-uri. Rezolvarea se va face maximizând concurența thread-urilor. Rezultatul se va printa de către un thread ales de către programator (poate fi thread-ul cu un anumit ID), dar **NU** de către thread-ul *main*. Mai mult, doar acestui thread îi va fi permis să citească valorile parțiale calculate de celelalte thread-uri.

**Atenție!** Thread-urile nu au voie să incrementeze niciun contor partajat de către toate, deoarece incrementarea lui presupune citirea valorii inițiale.

În urma compilării, va rezulta un executabil numit *testapd*, care se va rula astfel: `./testapd N P X` (unde X este valoarea căutată). Vectorul în care se caută valoarea va fi inițializat în funcția *main* ca  $v[i] = (i + 3) \% 5$ , înainte de pornirea thread-urilor. Programul va trebui să afișeze numărul de apariții ale elementului căutat în vector, apoi suma pozițiilor aparițiilor, cu spațiu între ele. Aveți mai jos trei exemple de rulare:

```
$ ./testapd 10 3 3
2 5

$ ./testapd 25 2 4
5 55

$ ./testapd 3 8 0
1 2
```

În primul exemplu de mai sus, vectorul este [3,4,0,1,2,3,4,0,1,2], deci elementul căutat (adică 3) se află la pozițiile 0 și 5. În acest caz, numărul de apariții este 2, iar suma acestora este 5.

**Atenție!** Toate thread-urile trebuie pornite în același timp (într-o buclă comună). Programul submis nu trebuie să printeze altceva decât numărul de apariții și suma pozițiilor aparițiilor, în această ordine și în formatul de mai sus. Afișarea se va face dintr-unul din thread-urile pornite de către voi.

**Atenție!** Formula cu care se inițializează vectorul este doar pentru a ușura testarea. Nu o folosiți pentru a deduce matematic numărul de apariții. Fiecare thread trebuie să itereze prin fragmentul de vector alocat lui și să găsească aparițiile valorii căutate.

Se presupune că parametrii de intrare sunt corecți (atât ca număr și ordine, cât și ca valori). Valorile acceptate pentru N, P și X sunt:  $0 < N \leq 10000$ ,  $0 < P \leq 8$ ,  $0 \leq X < 5$ . Programul trebuie să funcționeze corect pentru orice combinație de N, P și X din aceste intervale.

Trebuie să încărcați pe Moodle o arhivă care să conțină **în rădăcină** fișierele sursă, un Makefile cu target-ul *build* și eventual un fișier README.

**Atenție!** Fișierul README este necesar doar în cazul în care considerați că este nevoie de explicații adiționale în legătură cu implementarea realizată.