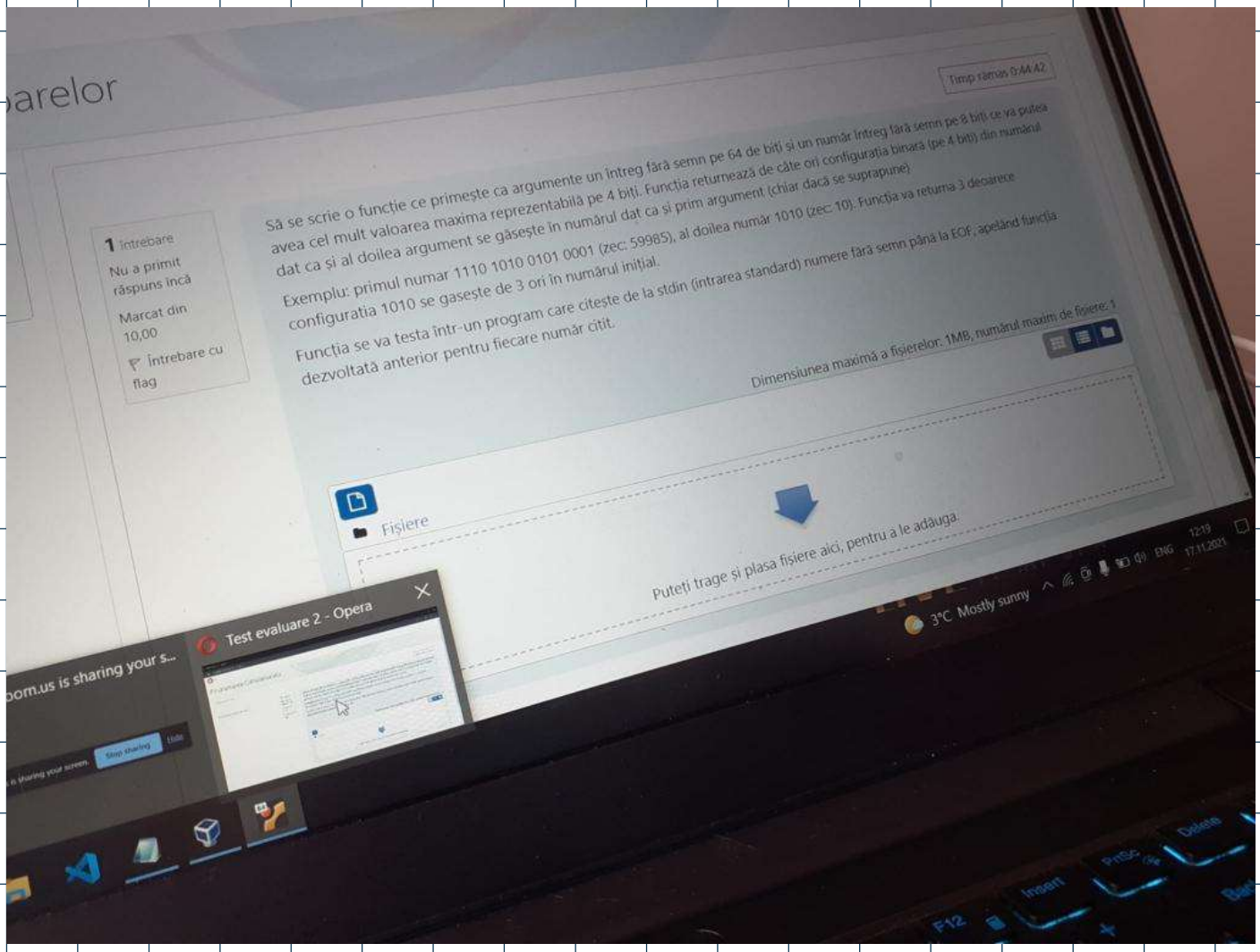
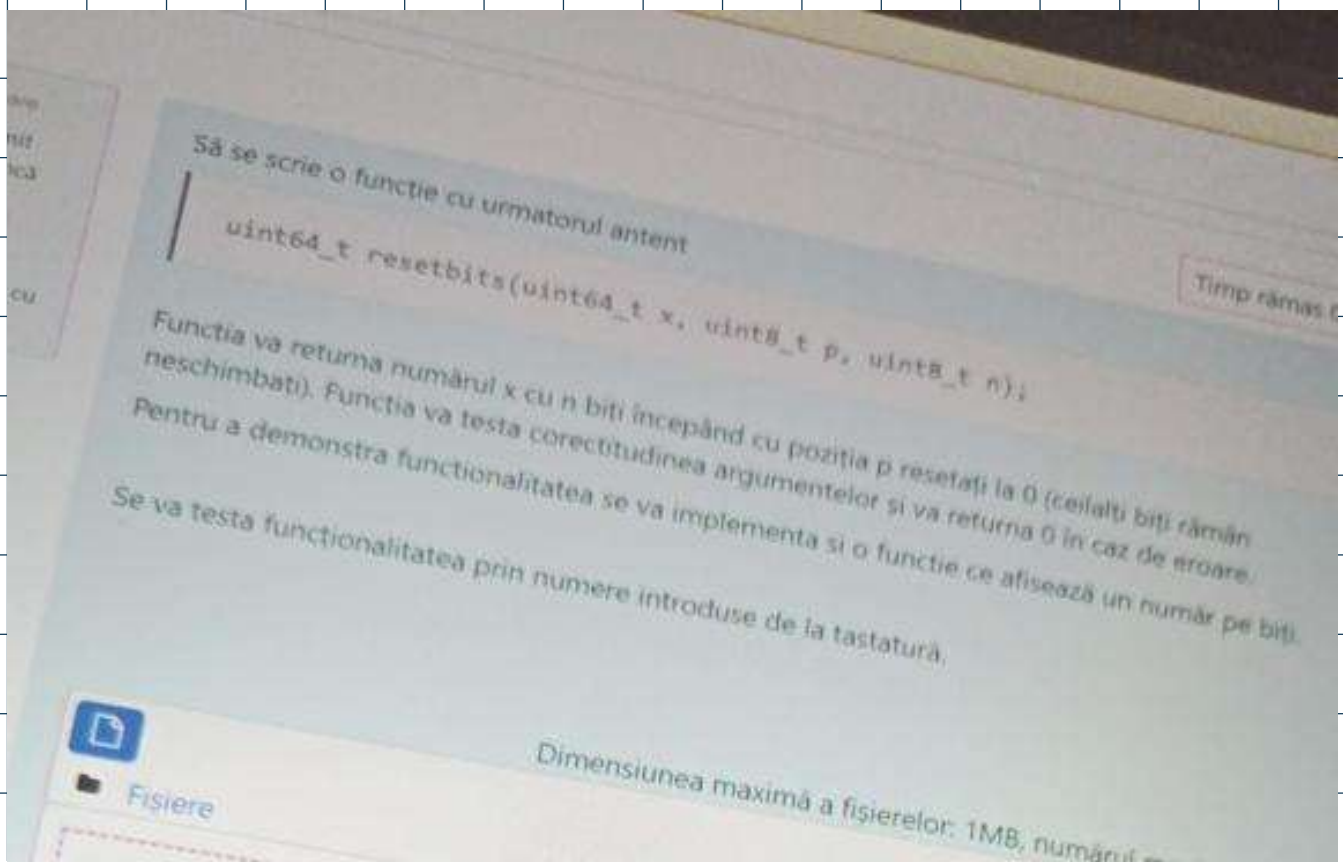
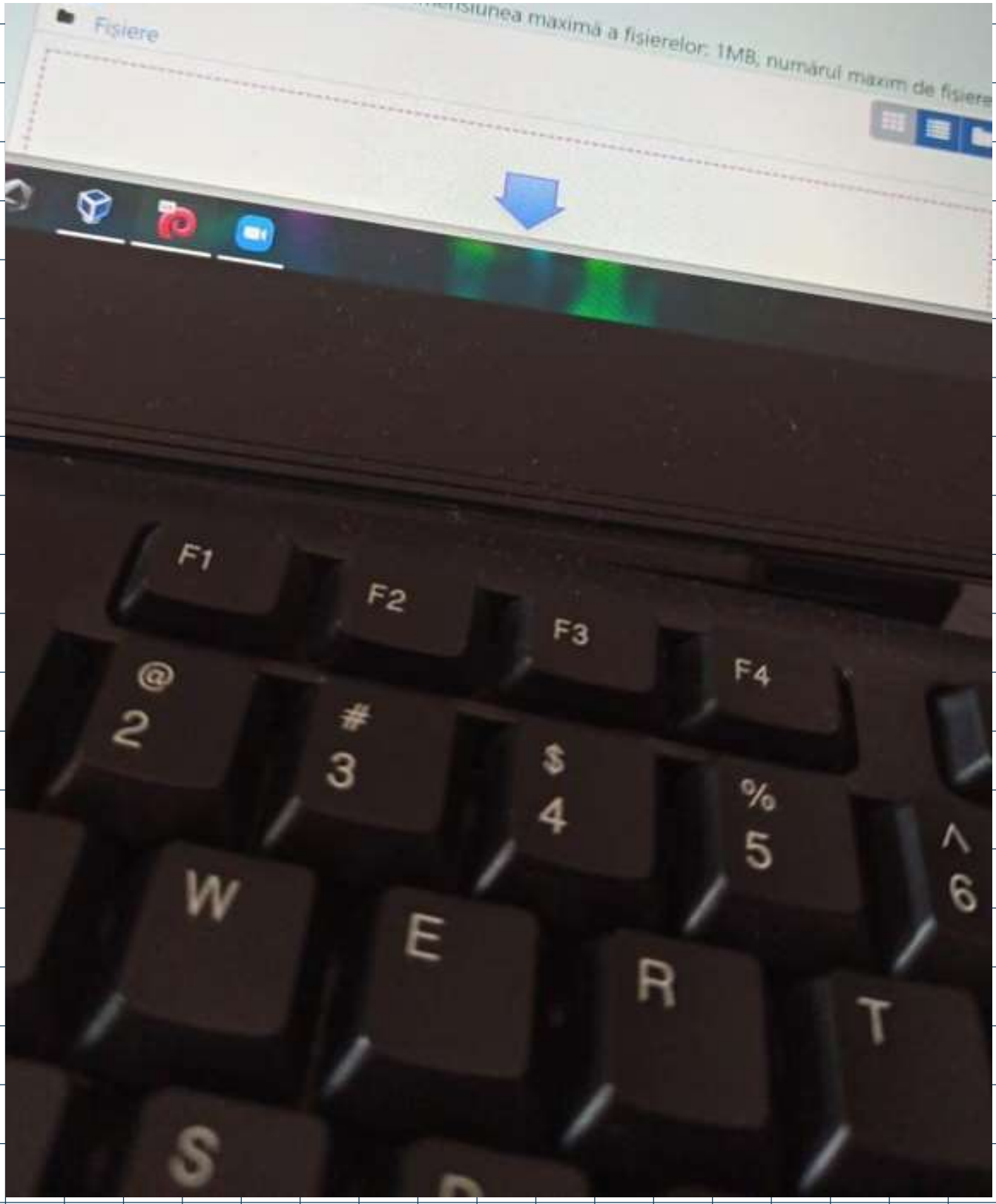
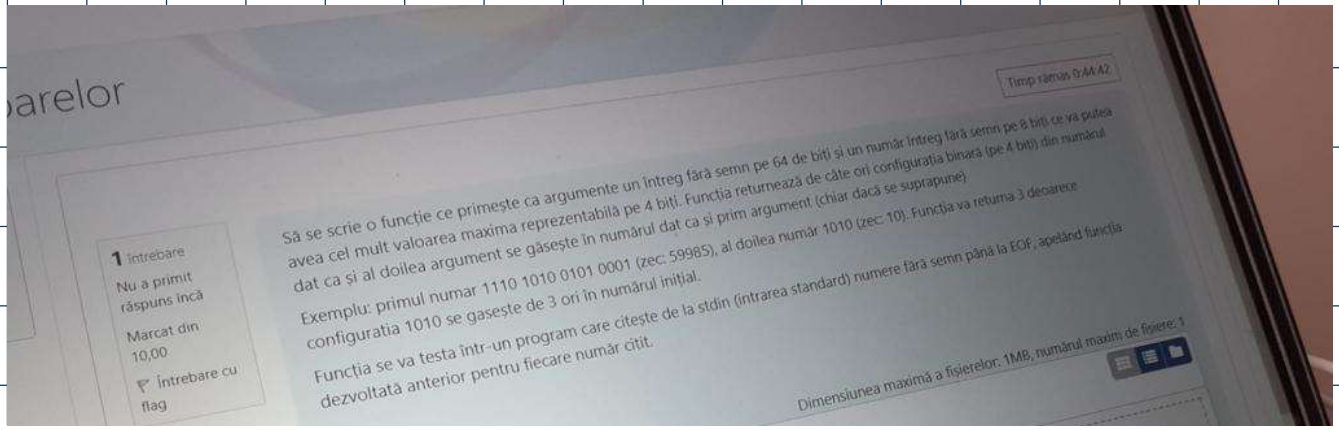
PROBLEME

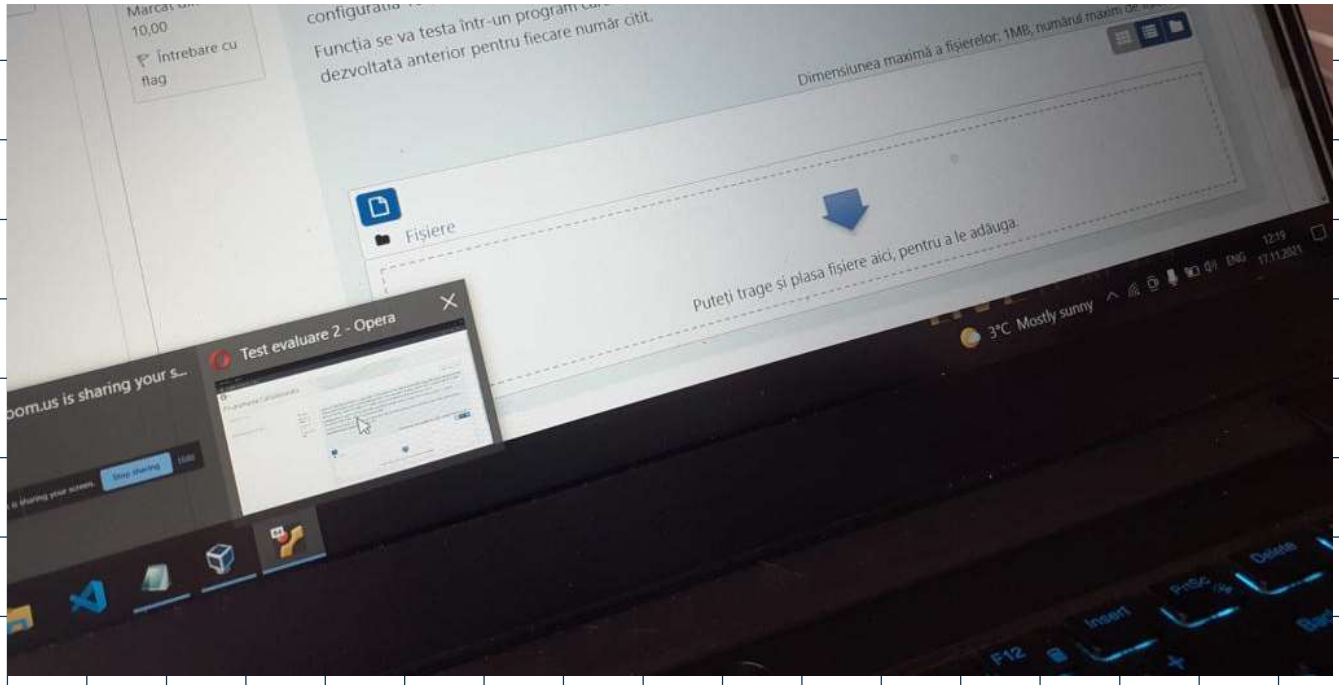
sâmbătă, 27 ianuarie 2024 15:51

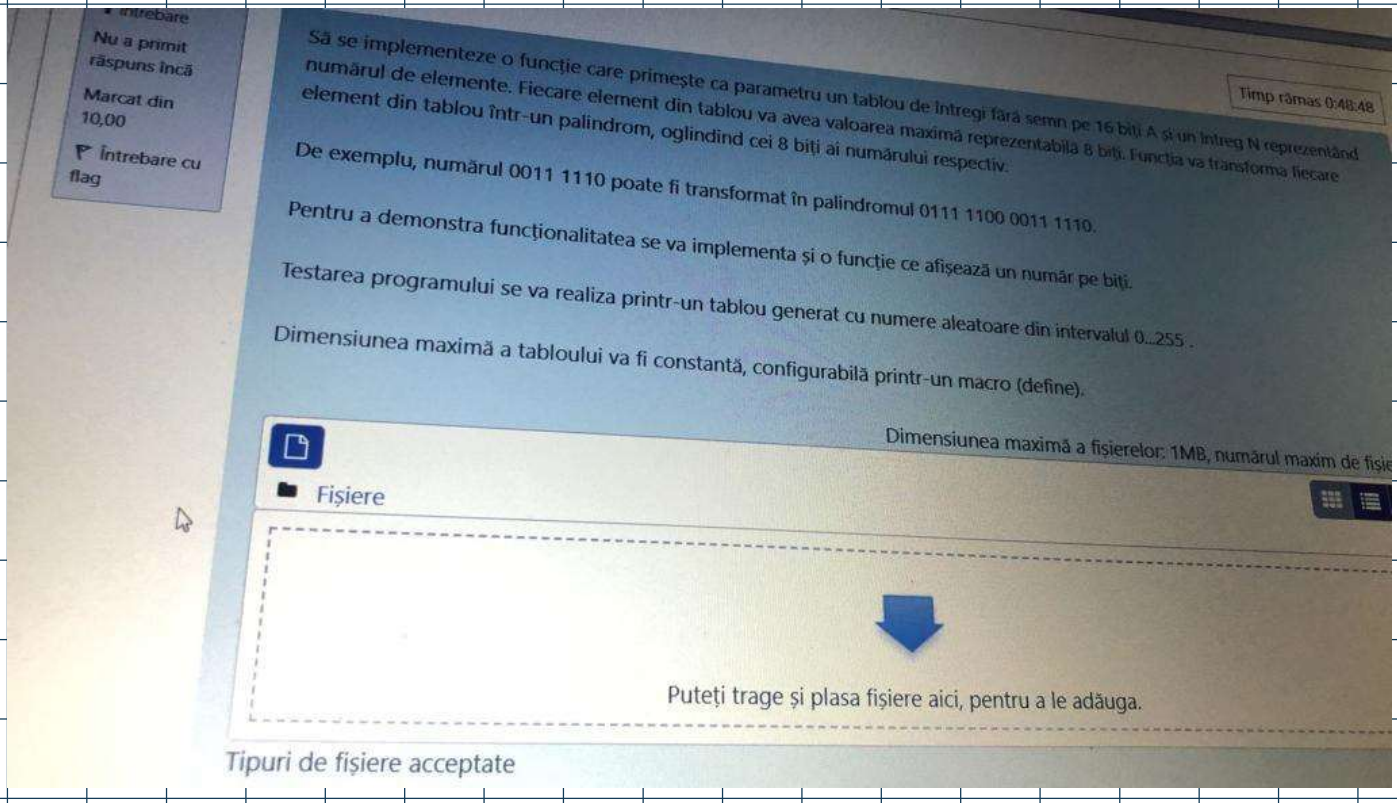


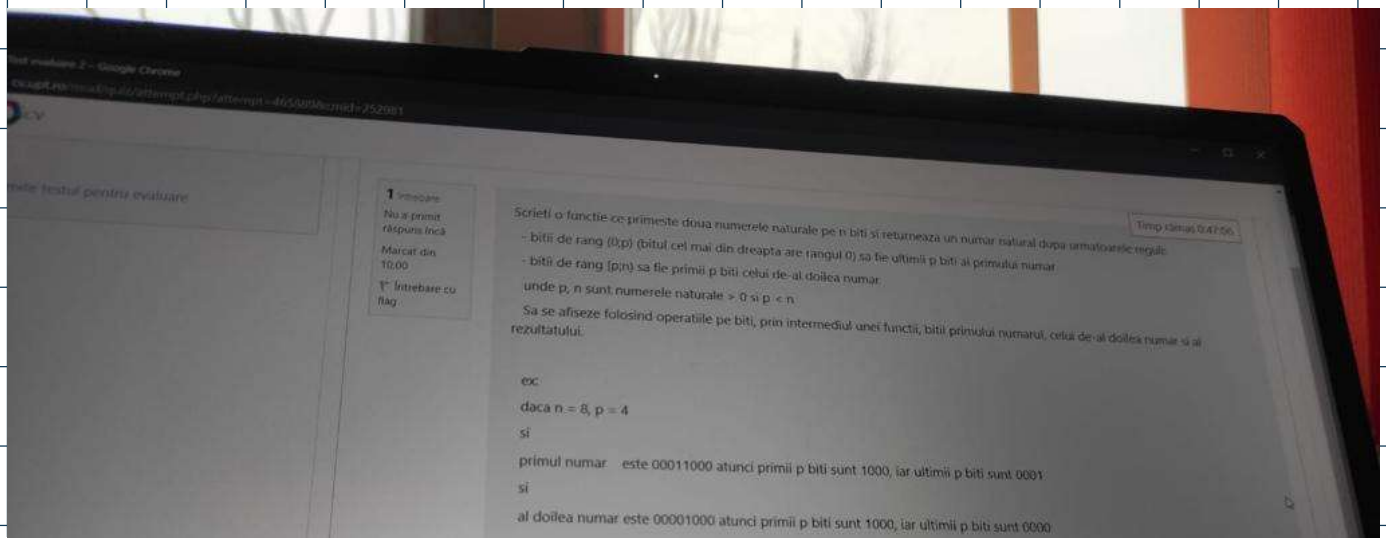



Fișiere
F1
F2
F3
F4
@
2
#
3
$
4
%
5
^
6
W
E
R
T
S

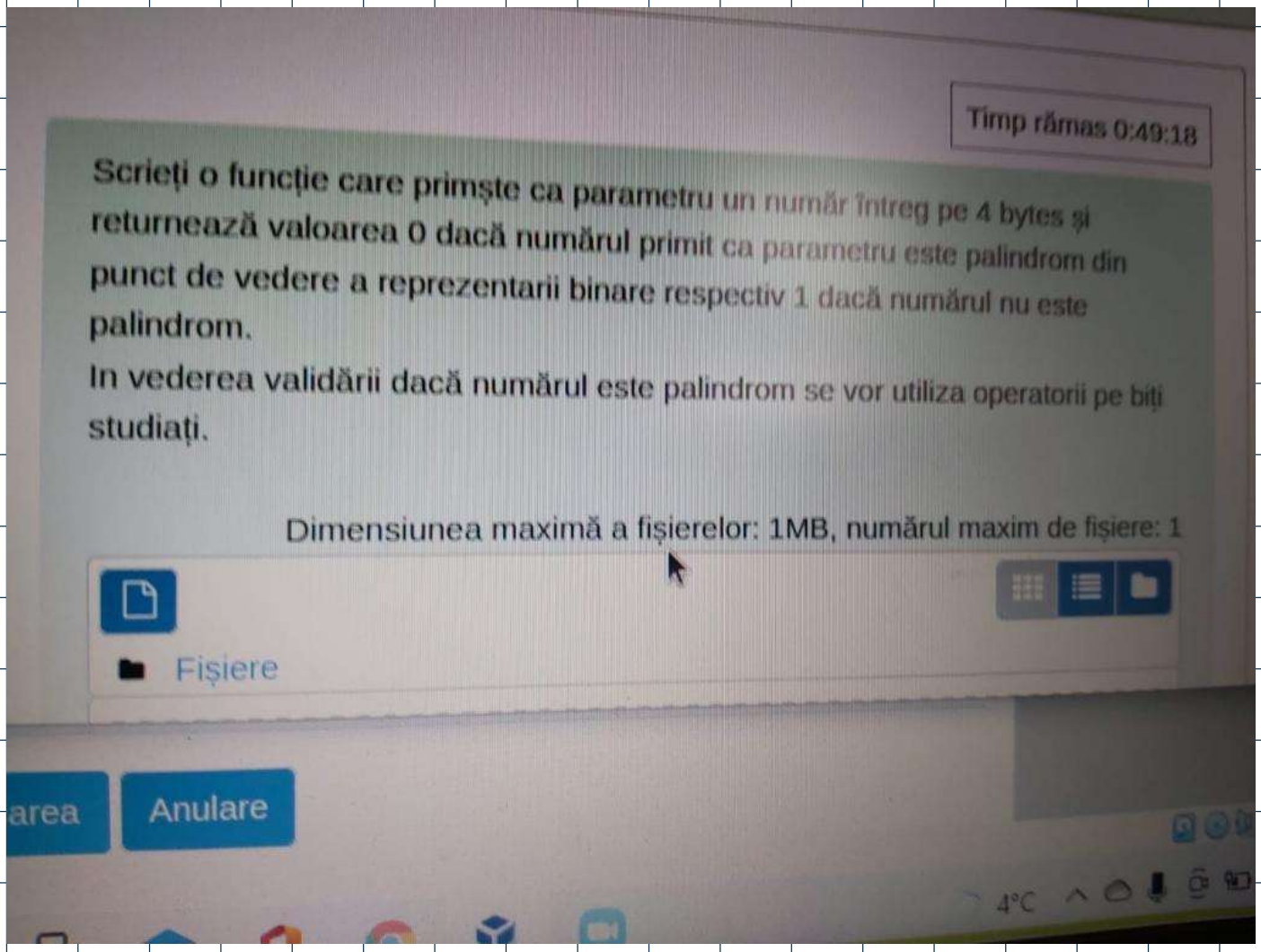

1 întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Să se scrie o funcție ce primește ca argumente un întreg fără semn pe 64 de biți și un număr întreg fără semn pe 8 biți ce va putea avea cel mult valoarea maximă reprezentabilă pe 4 biți. Funcția returnează de câte ori configurația binară (pe 4 biți) din numărul dat ca și al doilea argument se găsește în numărul dat ca și prim argument (chiar dacă se suprapune)
Exemplu: primul numar 1110 1010 0101 0001 (zec: 59985), al doilea număr 1010 (zec: 10). Funcția va returna 3 deoarece configuratia 1010 se gasește de 3 ori în numărul inițial.
Funcția se va testa într-un program care citește de la stdin (intrarea standard) numere fără semn până la EOF, apelând funcția dezvoltată anterior pentru fiecare număr citit.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

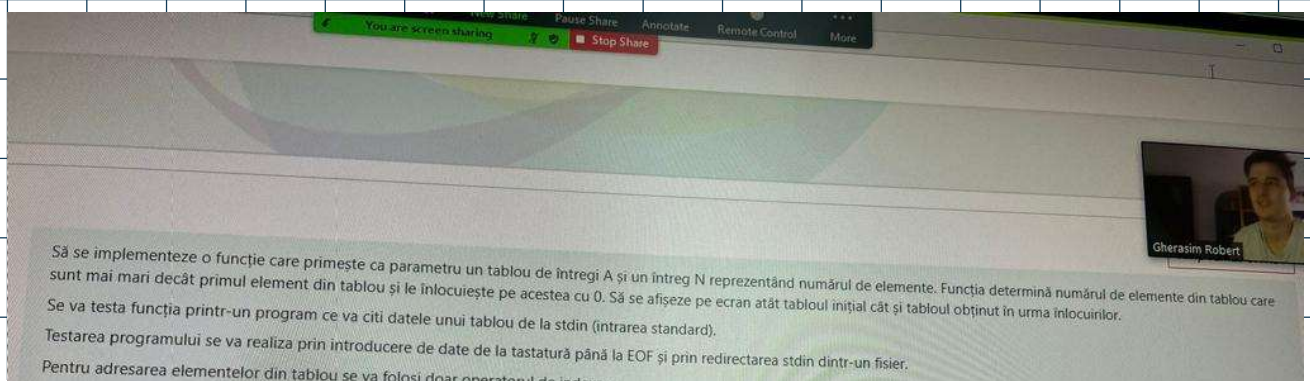

10,00
Întrebare cu flag
Funcția se va testa într-un program
dezvoltată anterior pentru fiecare număr citit.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Test evaluare 2 - Opera
3°C Mostly sunny

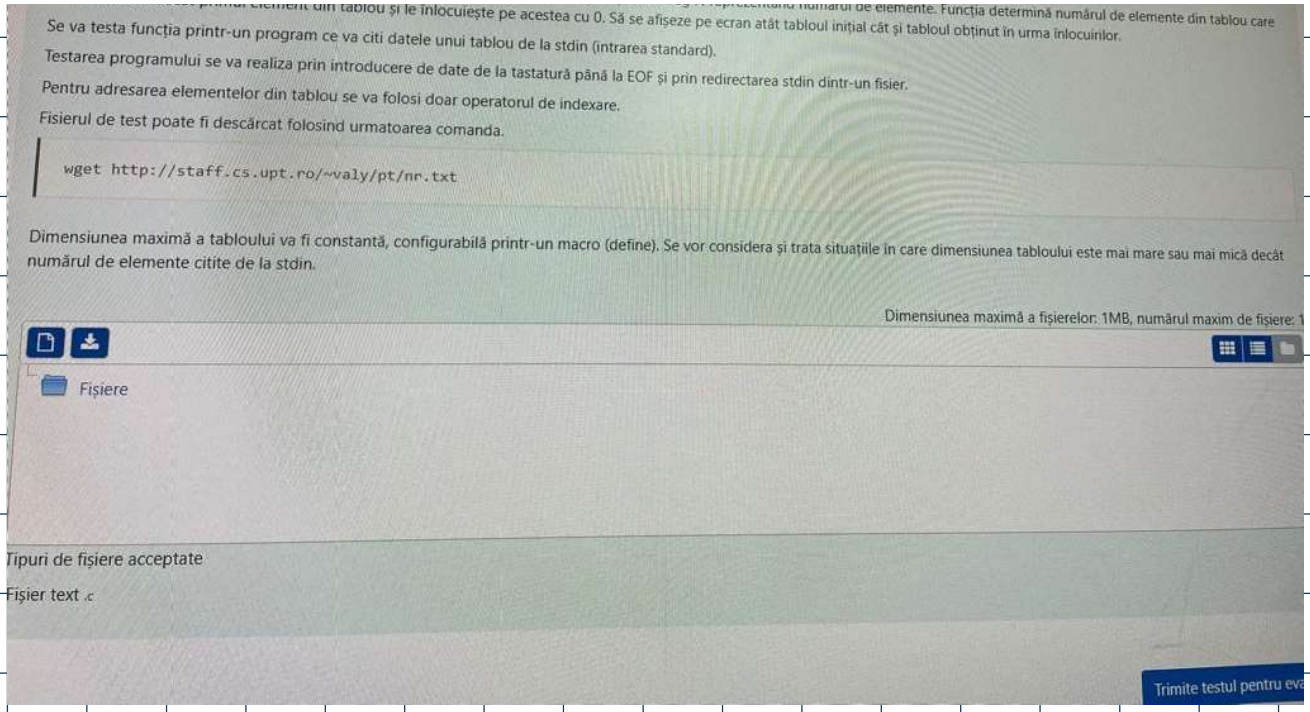

Întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Timp rămas 0:48:48
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 16 biți A și un întreg N reprezentând numărul de elemente. Fiecare element din tablou va avea valoarea maximă reprezentabilă 8 biți. Funcția va transforma fiecare element din tablou într-un palindrom, oglindind cei 8 biți ai numărului respectiv.
De exemplu, numărul 0011 1110 poate fi transformat în palindromul 0111 1100 0011 1110.
Pentru a demonstra funcționalitatea se va implementa și o funcție ce afișează un număr pe biți.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...255 .
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișie
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Tipuri de fișiere acceptate

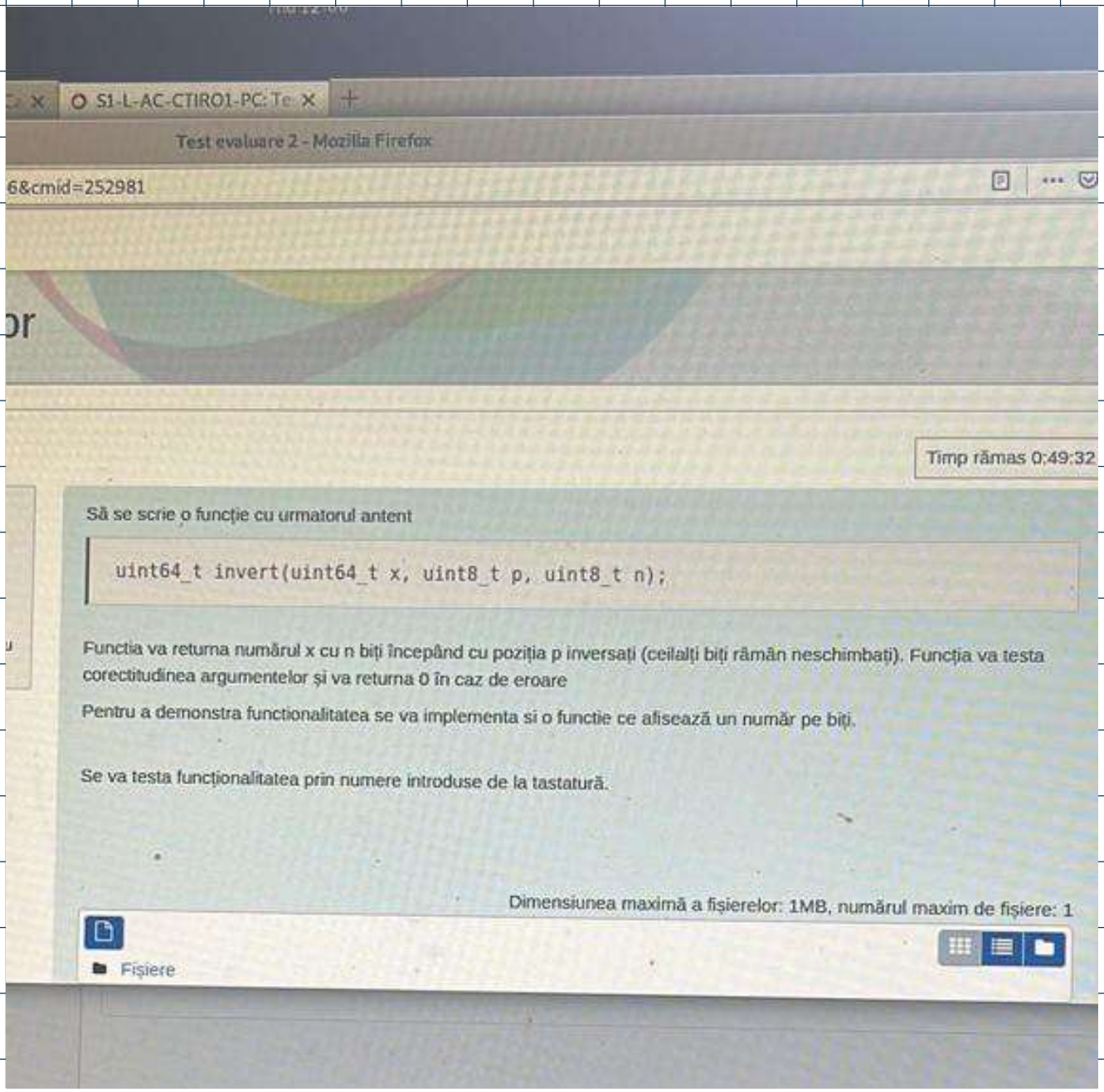

1 Întrebare
Nu a primit răspuns încă
Marcat din 10.00
Întrebare cu flag
Timp rămas 0:47:56
Scrieți o funcție ce primește două numerele naturale pe n biți și returnează un număr natural după următoarele reguli:
- biții de rang (0;p) (bitul cel mai din dreapta are rangul 0) să fie ultimii p biți ai primului număr
- biții de rang [p;n) să fie primii p biți celui de-al doilea număr
unde p, n sunt numerele naturale > 0 si p < n
Sa se afiseze folosind operatiile pe biti, prin intermediul unei functii, bitii primului numarul, celui de-al doilea numar si al rezultatului.
ex:
daca n = 8, p = 4
si
primul numar este 00011000 atunci primii p biti sunt 1000, iar ultimii p biti sunt 0001
si
al doilea numar este 00001000 atunci primii p biti sunt 1000, iar ultimii p biti sunt 0000

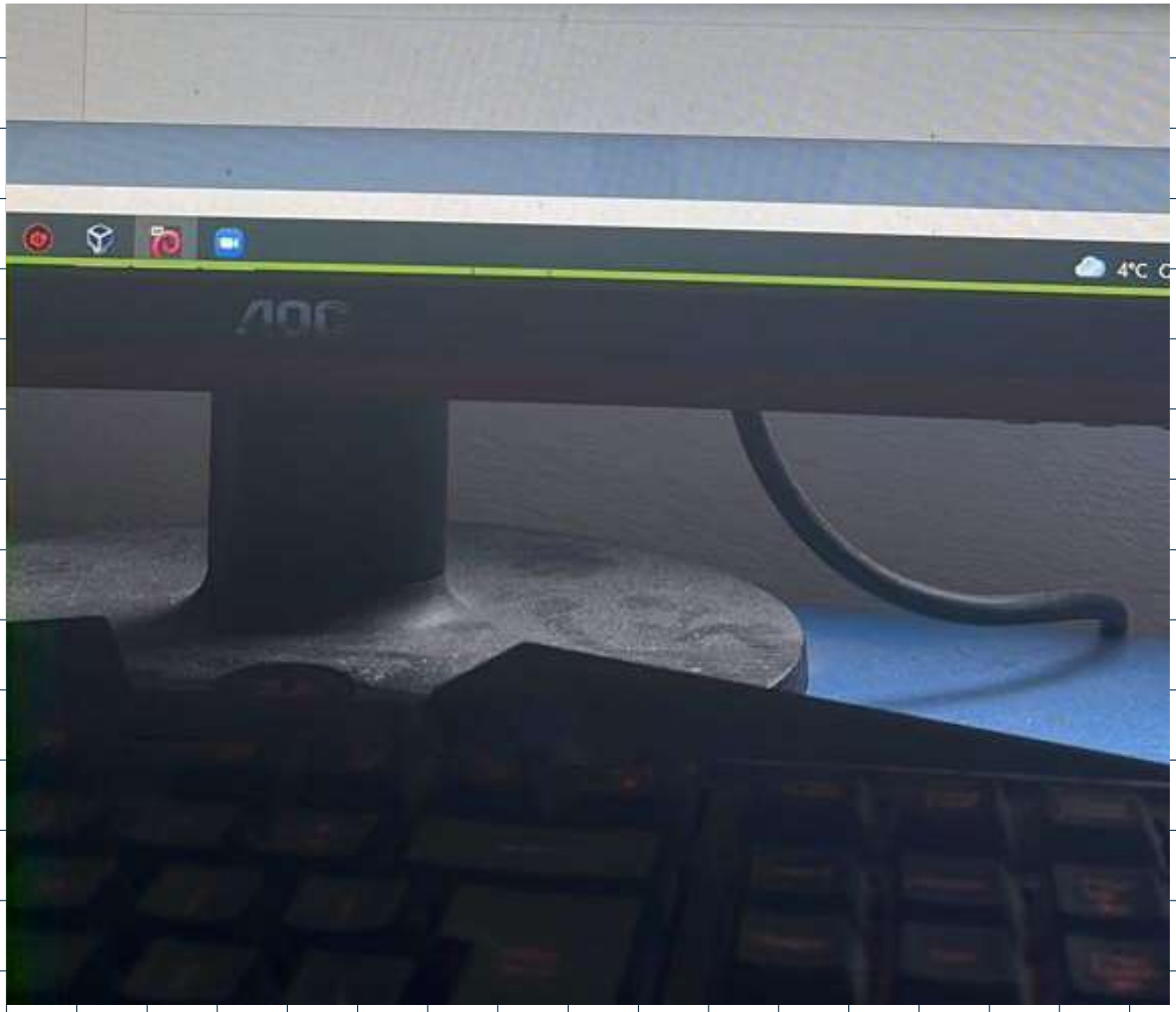

primul numar este 00011000 atunci primii p biti sunt 1000, iar ultimii p biti sunt 0001
si
al doilea numar este 00001000 atunci primii p biti sunt 1000, iar ultimii p biti sunt 0000
atunci rezultatul este 10000001
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
2°C Cloudy
ROU

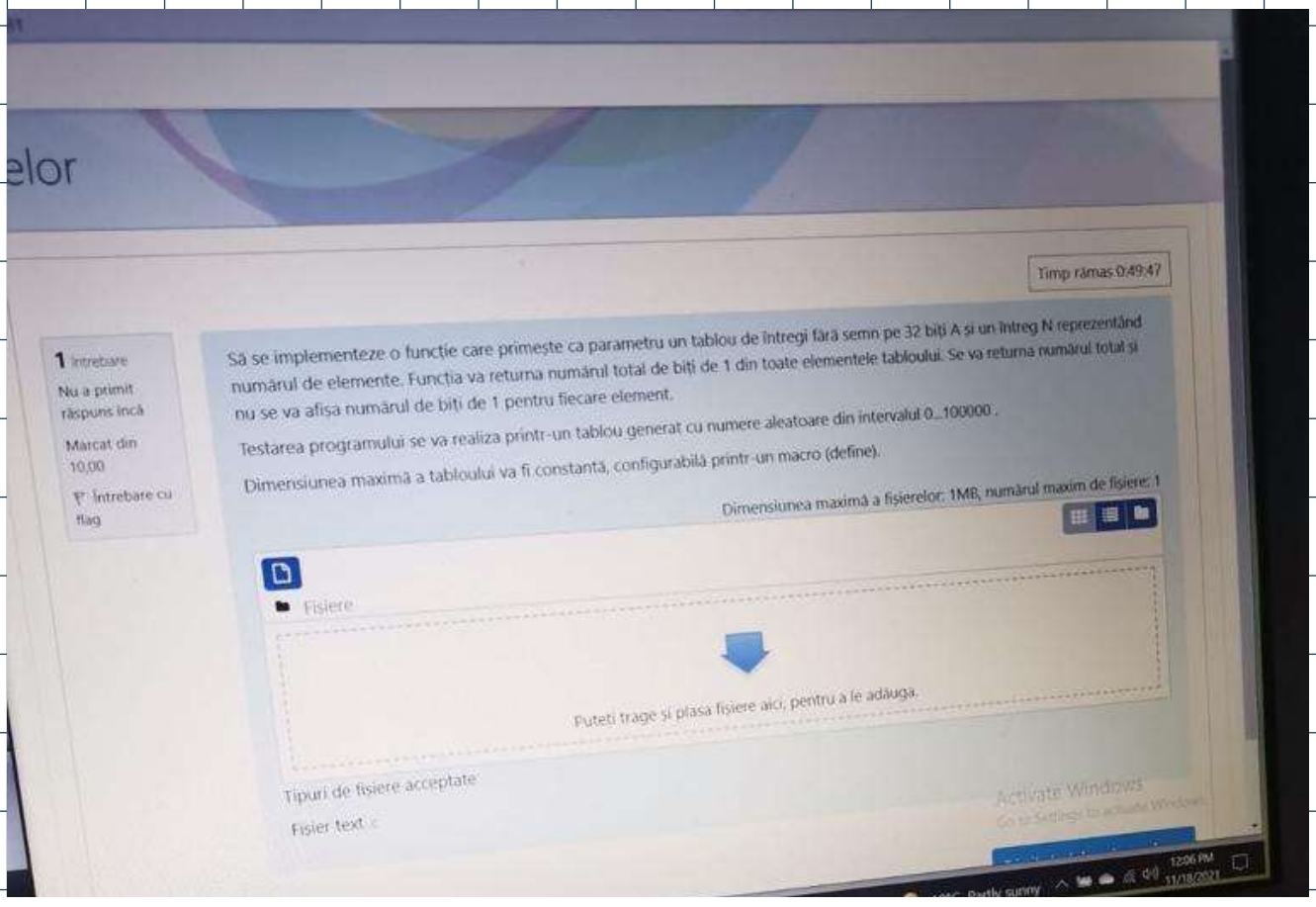

Timp rămas 0:49:18
Scrieți o funcție care primște ca parametru un număr întreg pe 4 bytes și returnează valoarea 0 dacă numărul primit ca parametru este palindrom din punct de vedere a reprezentarii binare respectiv 1 dacă numărul nu este palindrom.
In vederea validării dacă numărul este palindrom se vor utiliza operatorii pe biți studiați.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Anulare
4°C

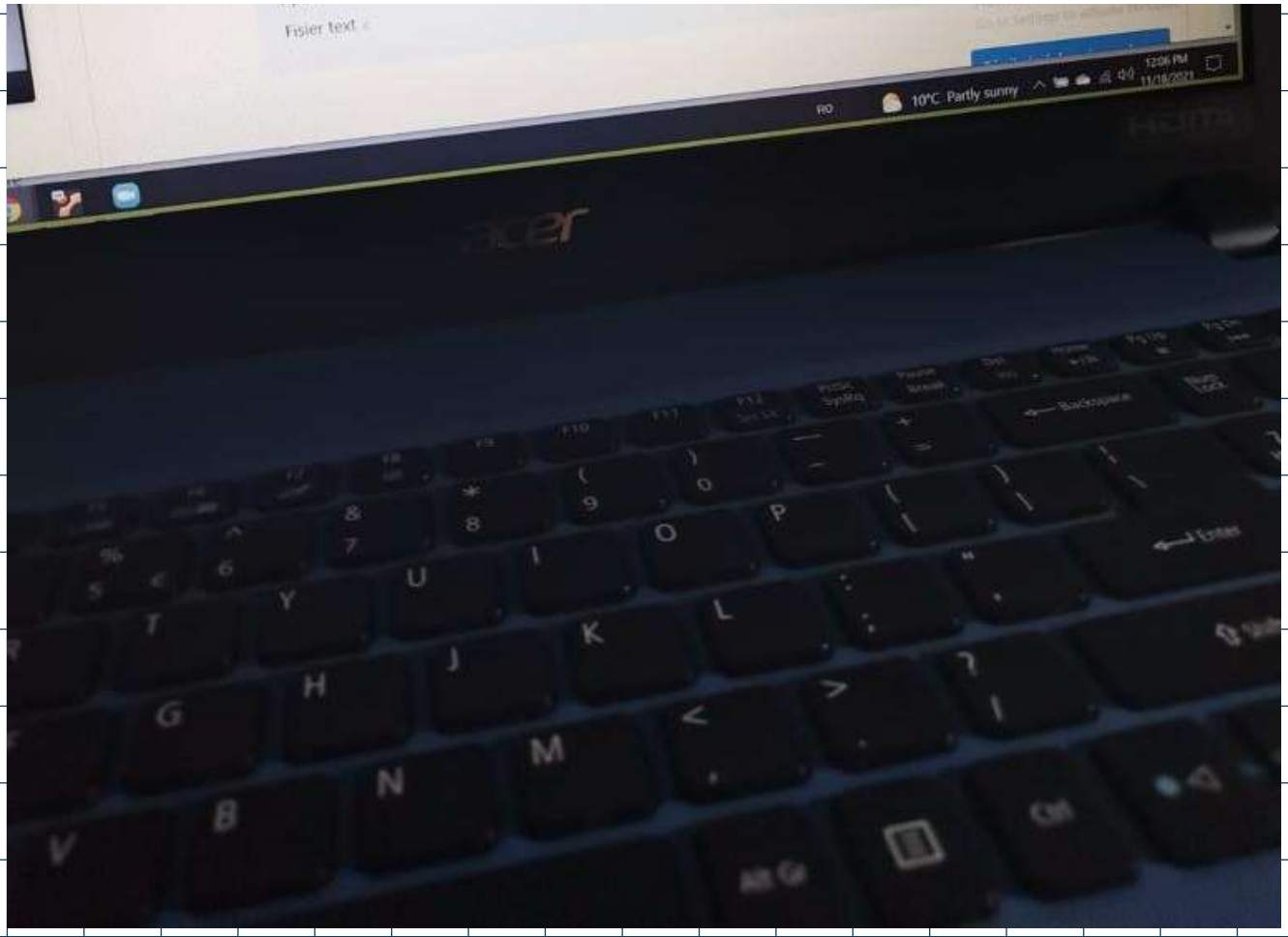

You are screen sharing
Stop Share
Pause Share
Annotate
Remote Control
More
Gherasim Robert
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un întreg N reprezentând numărul de elemente. Funcția determină numărul de elemente din tablou care sunt mai mari decât primul element din tablou și le înlocuiește pe acestea cu 0. Să se afișeze pe ecran atât tabloul inițial cât și tabloul obținut în urma înlocuirilor.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fisier.
Pentru adresarea elementelor din tablou se va folosi doar operatorul de

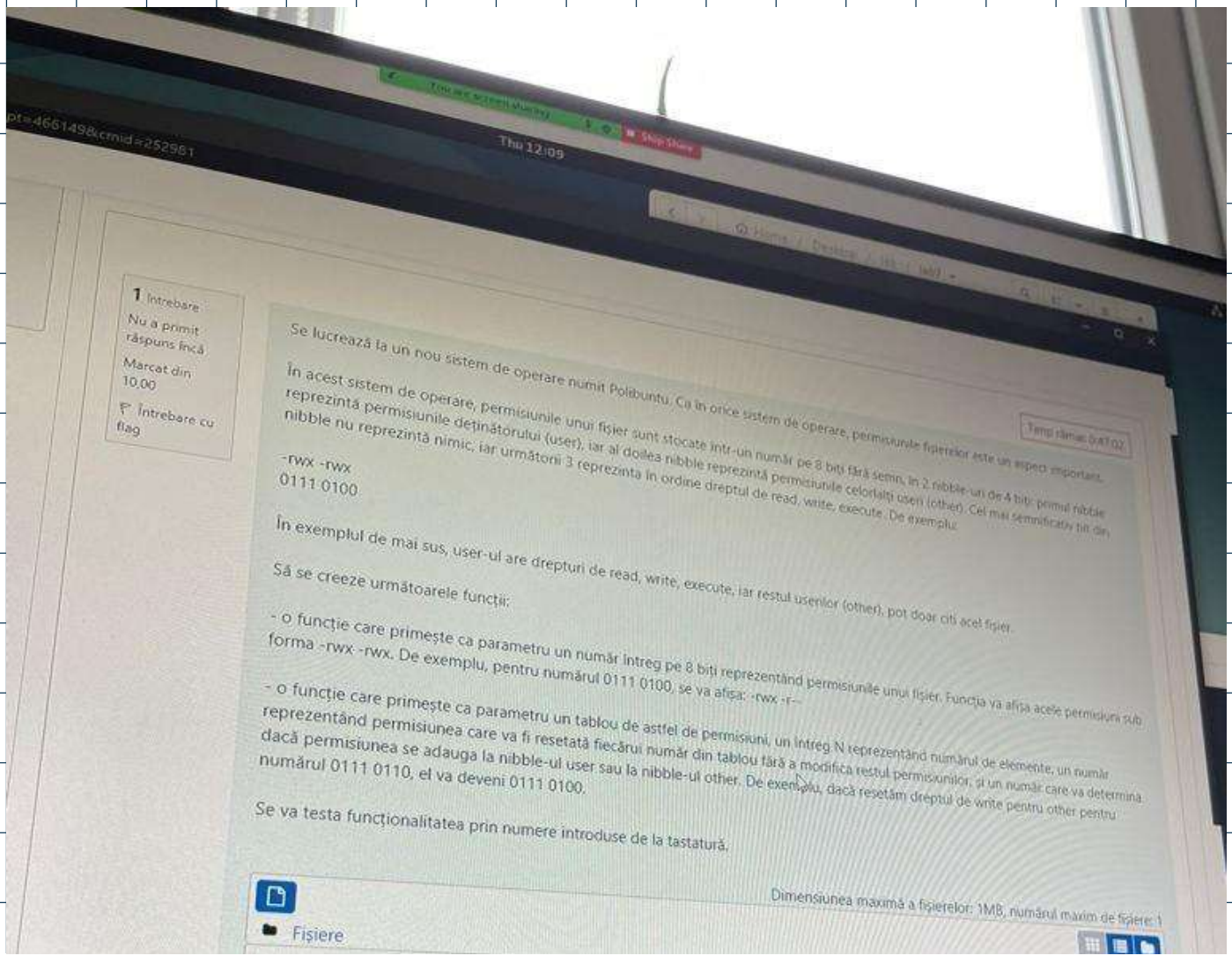

Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fișier.
Pentru adresarea elementelor din tablou se va folosi doar operatorul de indexare.
Fisierul de test poate fi descărcat folosind urmatoarea comanda.
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere:
Fișiere
Tipuri de fișiere acceptate
Fișier text .c
Trimite testul pentru

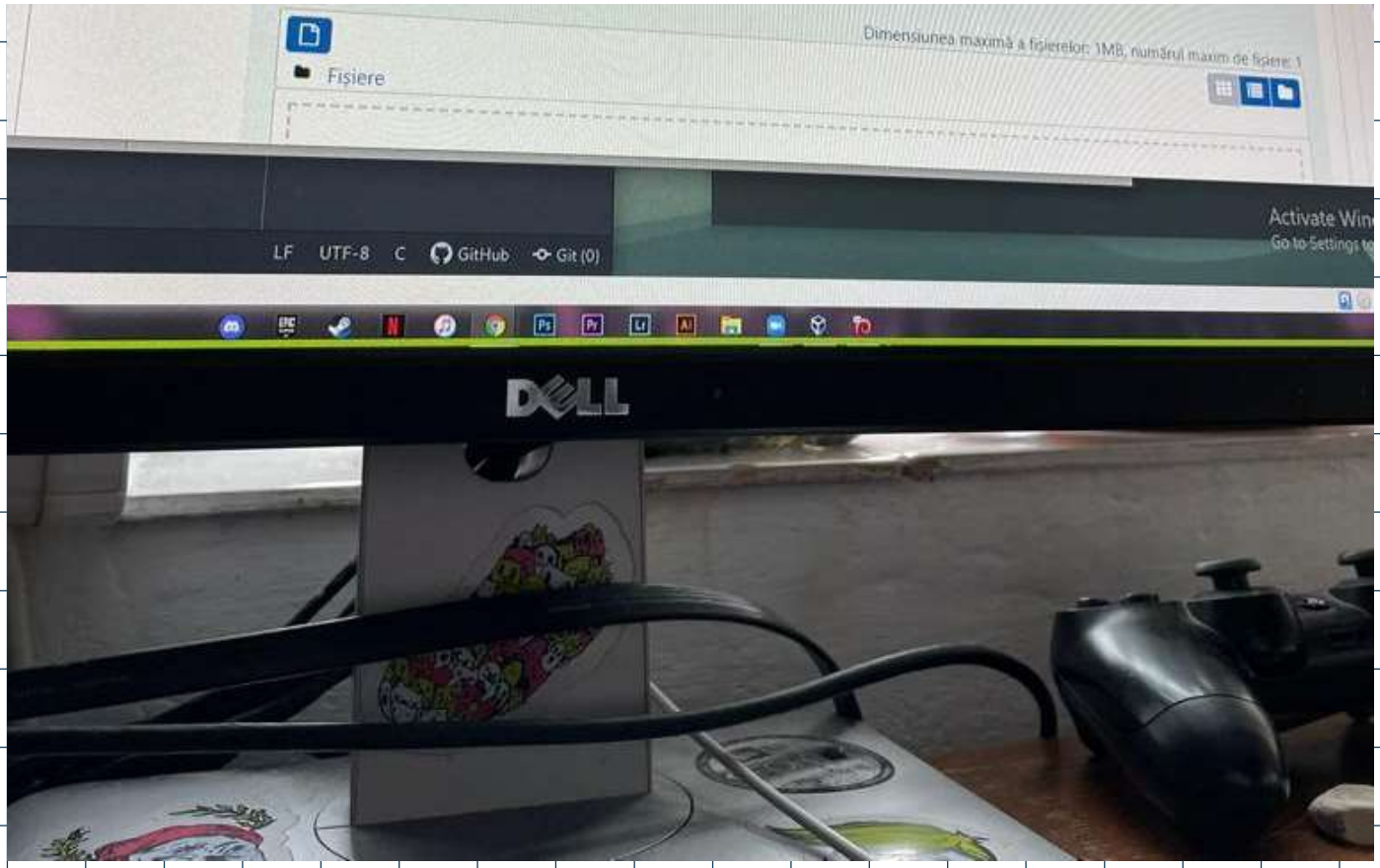

S1-L-AC-CTIRO1-PC: Te
Test evaluare 2 - Mozilla Firefox
6&cmid=252981
Timp rămas 0:49:32
Să se scrie o funcție cu urmatorul antent
uint64_t invert(uint64_t x, uint8_t p, uint8_t n);
Functia va returna numărul x cu n biți începând cu poziția p inversați (ceilalți biți rămân neschimbați). Funcția va testa corectitudinea argumentelor și va returna 0 în caz de eroare
Pentru a demonstra functionalitatea se va implementa si o functie ce afisează un număr pe biți.
Se va testa funcționalitatea prin numere introduse de la tastatură.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere

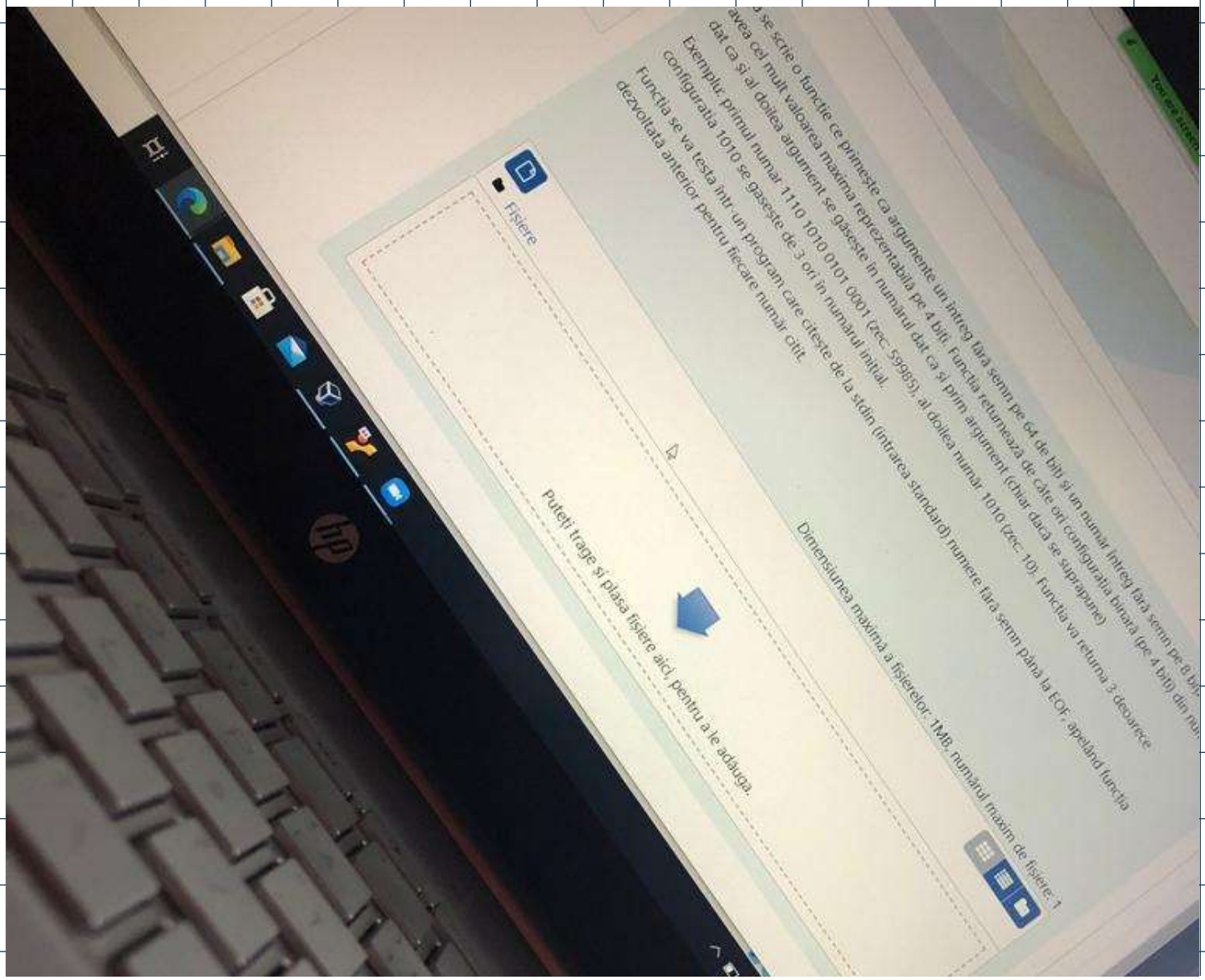

4°C


elor
Timp rămas 0:49:47
1 Întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 32 biți A și un întreg N reprezentând numărul de elemente. Funcția va returna numărul total de biți de 1 din toate elementele tabloului. Se va returna numărul total și nu se va afișa numărul de biți de 1 pentru fiecare element.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0..100000 .
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Tipuri de fișiere acceptate:
Fișier text
Activate Windows

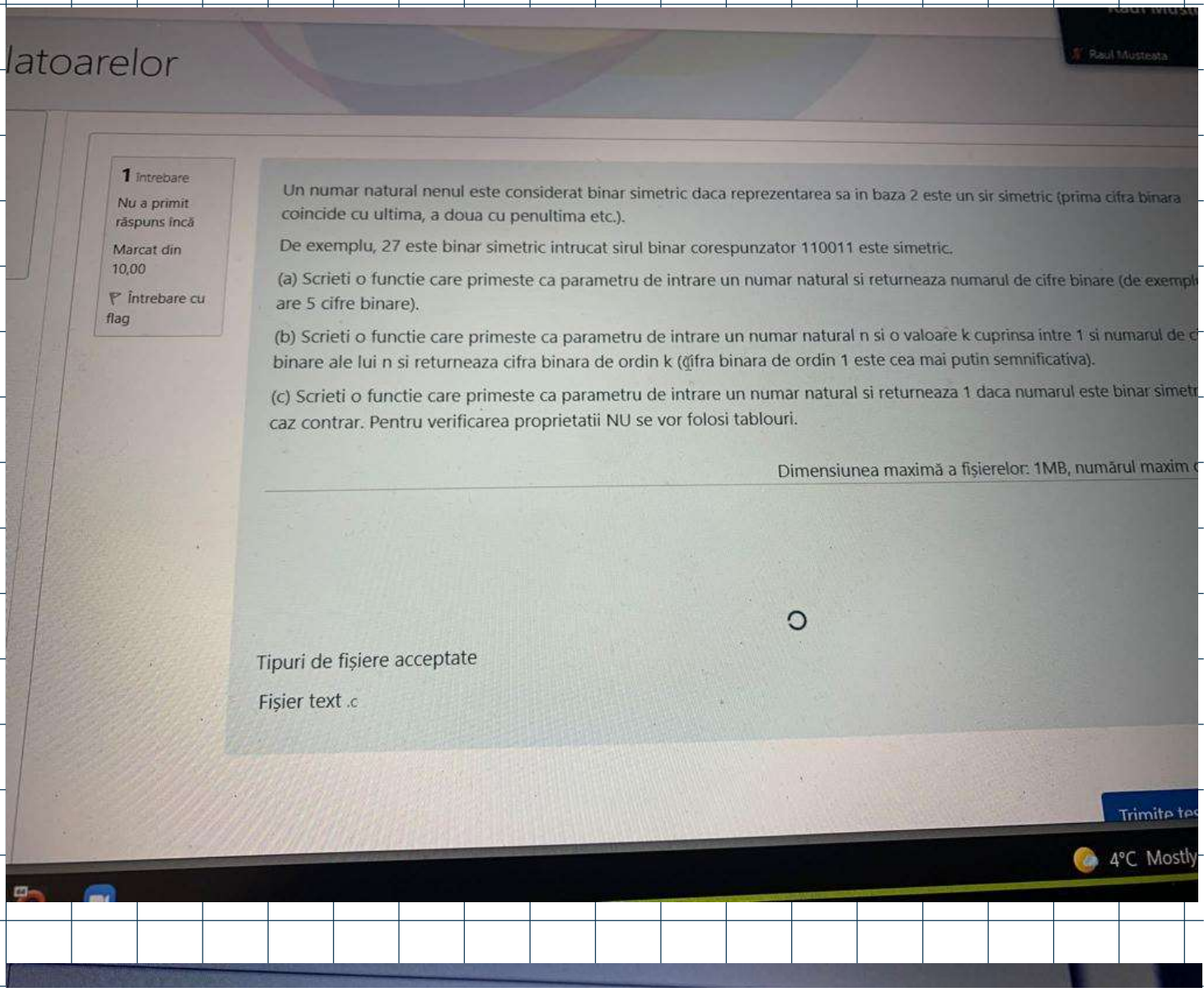

Fișier text
10°C Partly sunny
acer

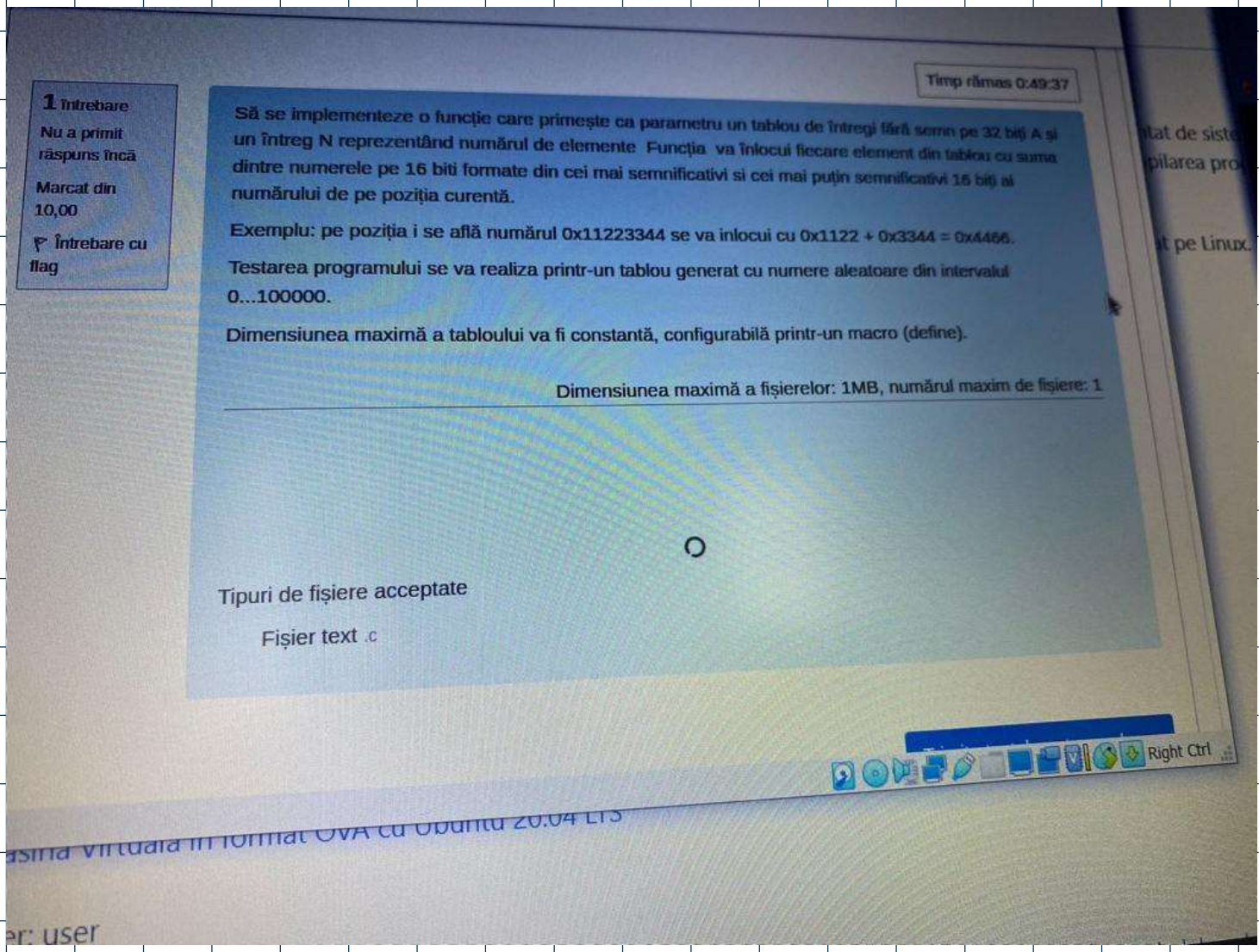

Thu 12:09
1 Întrebare
Nu a primit răspuns încă
Marcat din 10.00
Întrebare cu flag
Se lucrează la un nou sistem de operare numit Polibuntu. Ca în orice sistem de operare, permisiunile fișierelor este un aspect important.
În acest sistem de operare, permisiunile unui fișier sunt stocate într-un număr pe 8 biți fără semn, în 2 nibble-uri de 4 biți: primul nibble reprezintă permisiunile deținătorului (user), iar al doilea nibble reprezintă permisiunile celorlalți useri (other). Cel mai semnificativ bit din nibble nu reprezintă nimic, iar următorii 3 reprezintă în ordine dreptul de read, write, execute. De exemplu:
-rwx -rwx
0111 0100
În exemplul de mai sus, user-ul are drepturi de read, write, execute, iar restul userilor (other), pot doar citi acel fișier.
Să se creeze următoarele funcții:
- o funcție care primește ca parametru un număr întreg pe 8 biți reprezentând permisiunile unui fișier. Funcția va afișa acele permisiuni sub forma -rwx -rwx. De exemplu, pentru numărul 0111 0100, se va afișa: -rwx -r--
- o funcție care primește ca parametru un tablou de astfel de permisiuni, un întreg N reprezentând numărul de elemente, un număr reprezentând permisiunea care va fi resetată fiecărui număr din tablou fără a modifica restul permisiunilor, și un număr care va determina dacă permisiunea se adaugă la nibble-ul user sau la nibble-ul other. De exemplu, dacă resetăm dreptul de write pentru other pentru numărul 0111 0110, el va deveni 0111 0100.
Se va testa funcționalitatea prin numere introduse de la tastatură.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere

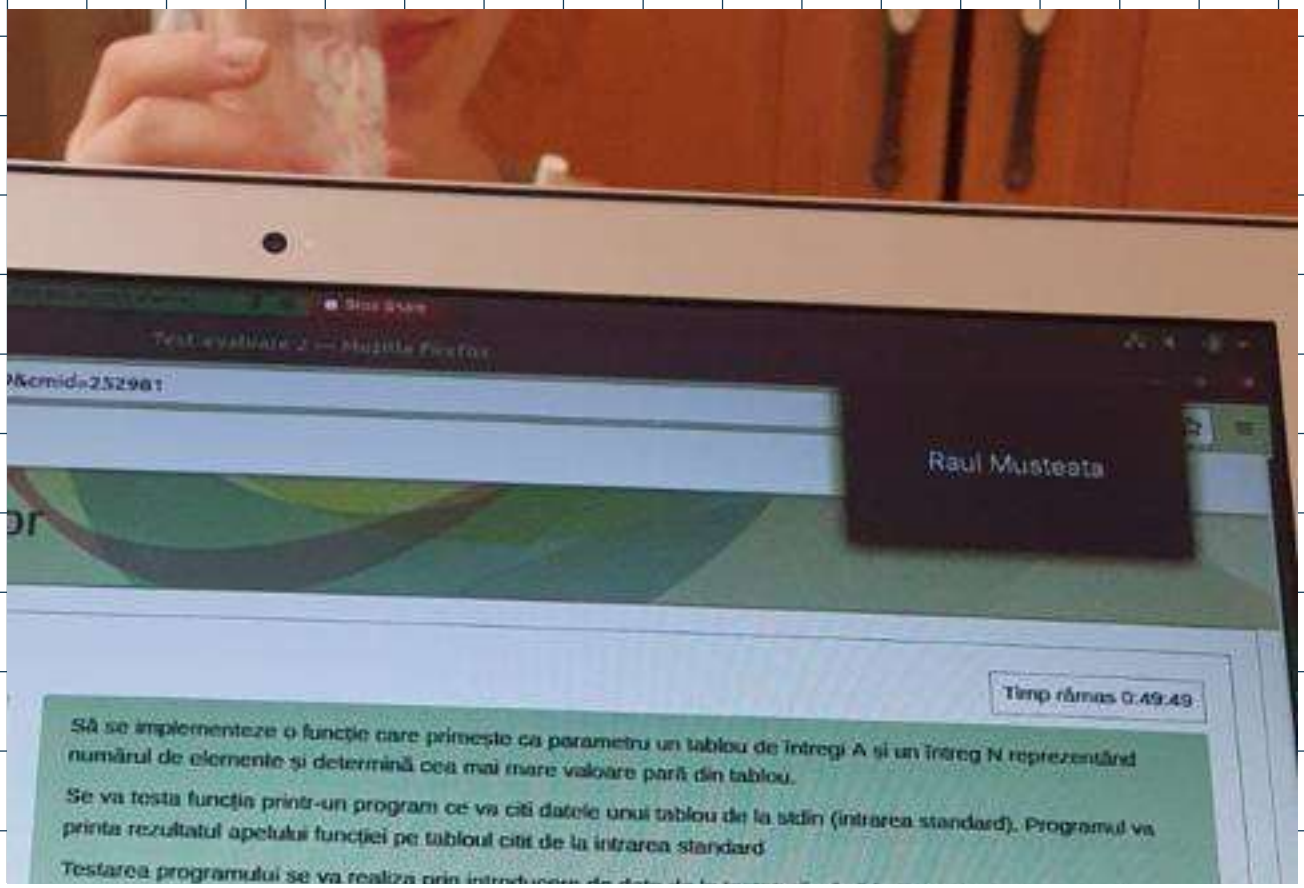

Fisiere
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
LF UTF-8 C GitHub Git (0)
DELL


Fisiere
Puteți trage și plasa fișiere aici pentru a le adăuga.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1


ROU
US
12:11
18.11.2021


latoarelor
Raul Musteata
1 Întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Un numar natural nenul este considerat binar simetric daca reprezentarea sa in baza 2 este un sir simetric (prima cifra binara coincide cu ultima, a doua cu penultima etc.).
De exemplu, 27 este binar simetric intrucat sirul binar corespunzator 110011 este simetric.
(a) Scrieti o functie care primeste ca parametru de intrare un numar natural si returneaza numarul de cifre binare (de exemplu are 5 cifre binare).
(b) Scrieti o functie care primeste ca parametru de intrare un numar natural n si o valoare k cuprinsa intre 1 si numarul de cifre binare ale lui n si returneaza cifra binara de ordin k (cifra binara de ordin 1 este cea mai putin semnificativa).
(c) Scrieti o functie care primeste ca parametru de intrare un numar natural si returneaza 1 daca numarul este binar simetric, 0 in caz contrar. Pentru verificarea proprietatii NU se vor folosi tablouri.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim
Tipuri de fișiere acceptate
Fișier text .c
Trimite tes
4°C Mostly


1 Întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Timp rămas 0:49:37
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 32 biți A și un întreg N reprezentând numărul de elemente. Funcția va înlocui fiecare element din tablou cu suma dintre numerele pe 16 biți formate din cei mai semnificativi și cei mai puțin semnificativi 16 biți ai numărului de pe poziția curentă.
Exemplu: pe poziția i se află numărul 0x11223344 se va inlocui cu 0x1122 + 0x3344 = 0x4466.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...100000.
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Tipuri de fișiere acceptate
Fișier text .c
Right Ctrl


Raul Musteata
Timp rămas 0:49:49
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un întreg N reprezentând numărul de elemente și determină cea mai mare valoare pară din tablou.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard). Programul va printa rezultatul apelului funcției pe tabloul citit de la intrarea standard
Testarea programului se va realiza prin

Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard), Programul va printa rezultatul apelului funcției pe tabloul citit de la intrarea standard

Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fișier.

Pentru adresarea elementelor din tablou se va folosi doar aritmetica cu pointeri si operatorul cu pointeri si nu se va permite utilizarea operatorului de indexare.

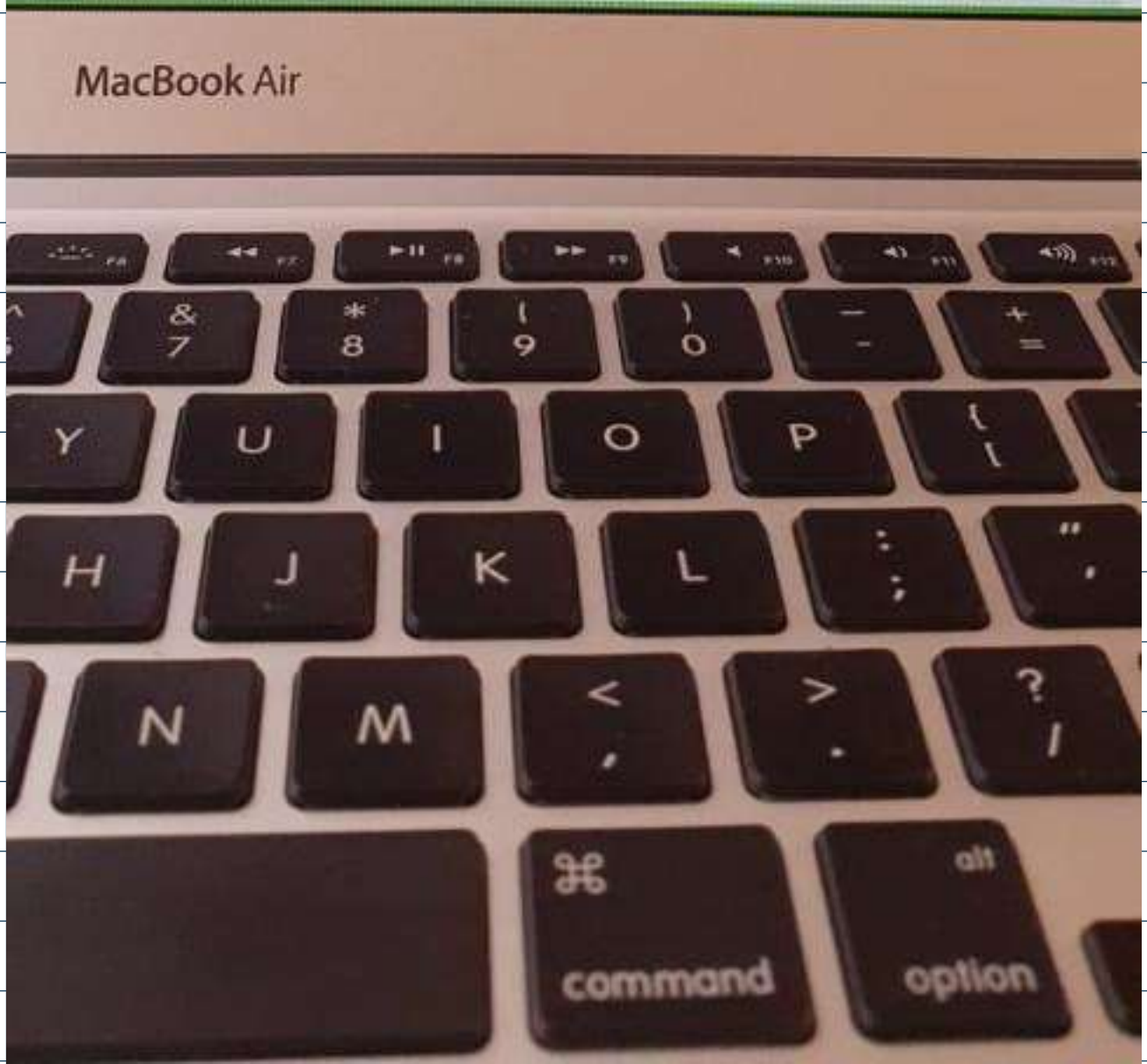
Fisierul de test poate fi descărcat folosind urmatoarea comanda.

```
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
```

Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

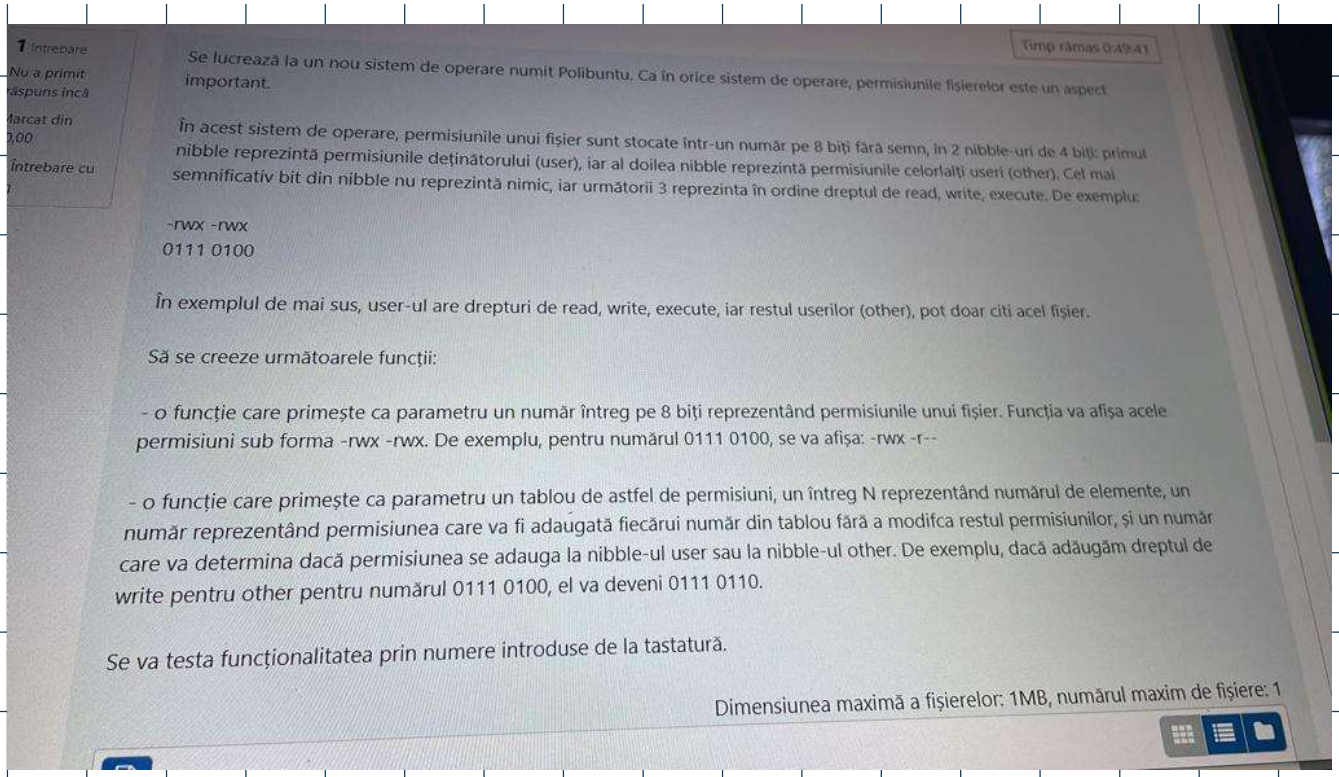
Tipuri de fișiere acceptate



1 Întrebare
Nu a primit răspuns încă

Timp rămas 0:49:41

Se lucrează la un nou sistem de operare numit Polibuntu. Ca în orice sistem de operare, permisiunile fișierelor este un aspect important.



1 întrebare
Nu a primit răspuns încă
Marcat din 2,00
Întrebare cu

Timp rămas 0:49:41

Se lucrează la un nou sistem de operare numit Polibuntu. Ca în orice sistem de operare, permisiunile fișierelor este un aspect important.

În acest sistem de operare, permisiunile unui fișier sunt stocate într-un număr pe 8 biți fără semn, în 2 nibble-uri de 4 biți: primul nibble reprezintă permisiunile deținătorului (user), iar al doilea nibble reprezintă permisiunile celorlalți useri (other). Cel mai semnificativ bit din nibble nu reprezintă nimic, iar următorii 3 reprezinta în ordine dreptul de read, write, execute. De exemplu:

-rwx -rwx
0111 0100

În exemplul de mai sus, user-ul are drepturi de read, write, execute, iar restul userilor (other), pot doar citi acel fișier.

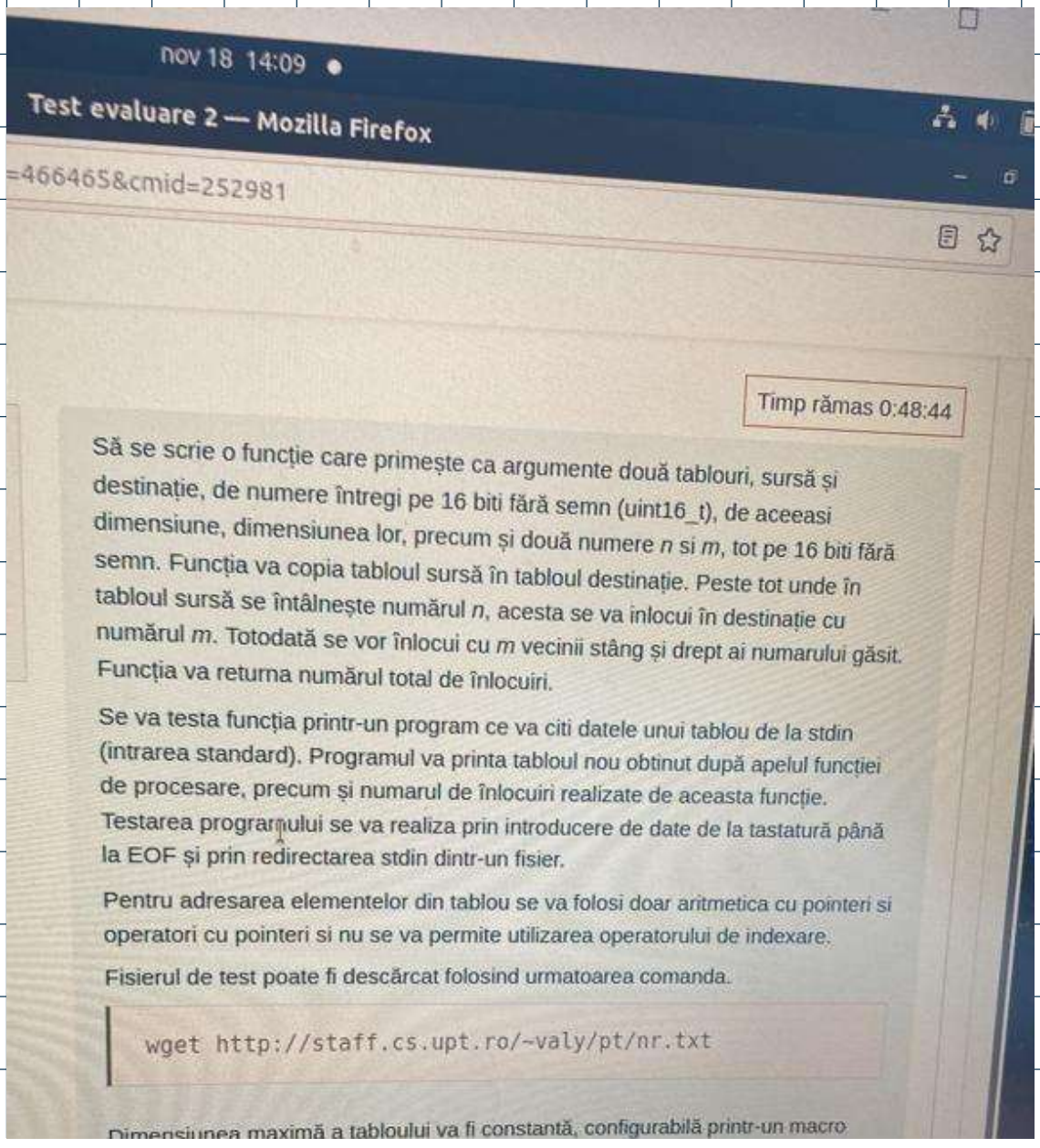
Să se creeze următoarele funcții:

- o funcție care primește ca parametru un număr întreg pe 8 biți reprezentând permisiunile unui fișier. Funcția va afișa acele permisiuni sub forma -rwx -rwx. De exemplu, pentru numărul 0111 0100, se va afișa: -rwx -r--

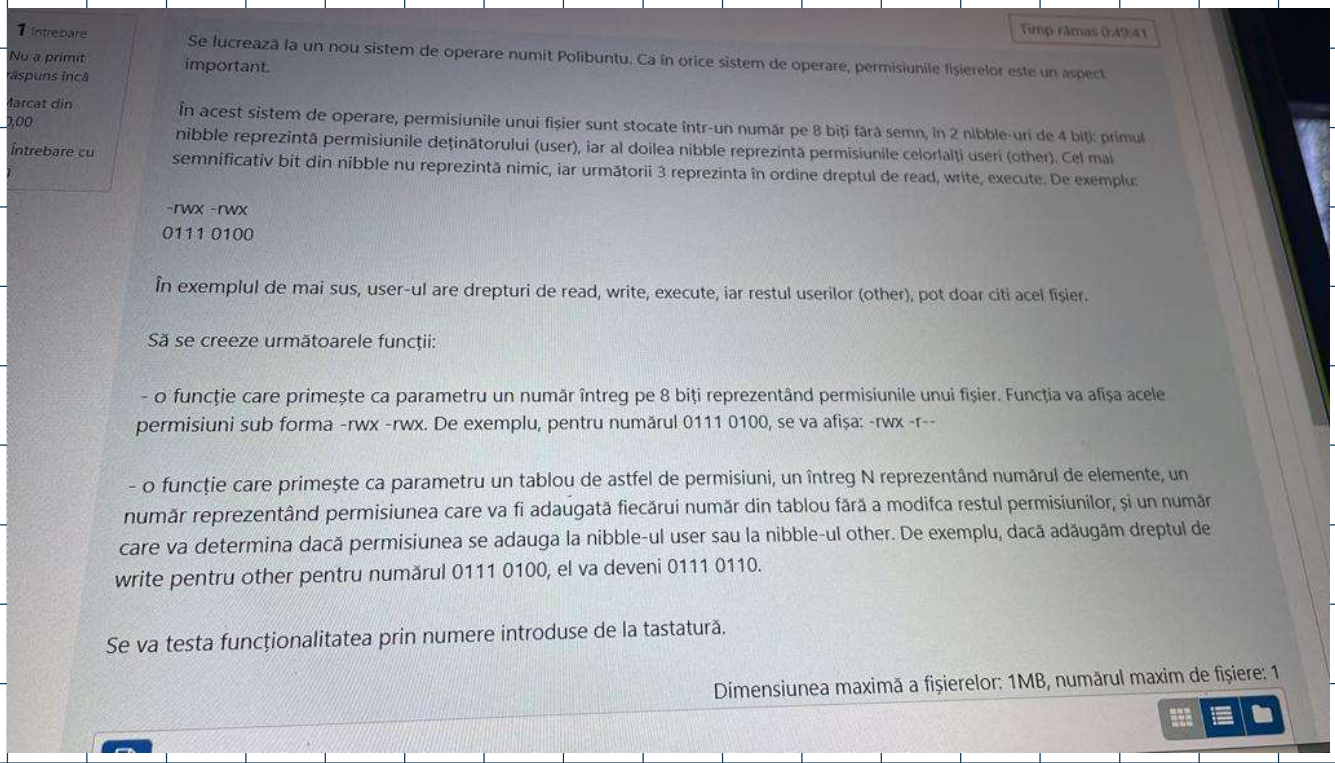
- o funcție care primește ca parametru un tablou de astfel de permisiuni, un întreg N reprezentând numărul de elemente, un număr reprezentând permisiunea care va fi adaugată fiecărui număr din tablou fără a modifca restul permisiunilor, și un număr care va determina dacă permisiunea se adauga la nibble-ul user sau la nibble-ul other. De exemplu, dacă adăugăm dreptul de write pentru other pentru numărul 0111 0100, el va deveni 0111 0110.

Se va testa funcționalitatea prin numere introduse de la tastatură.

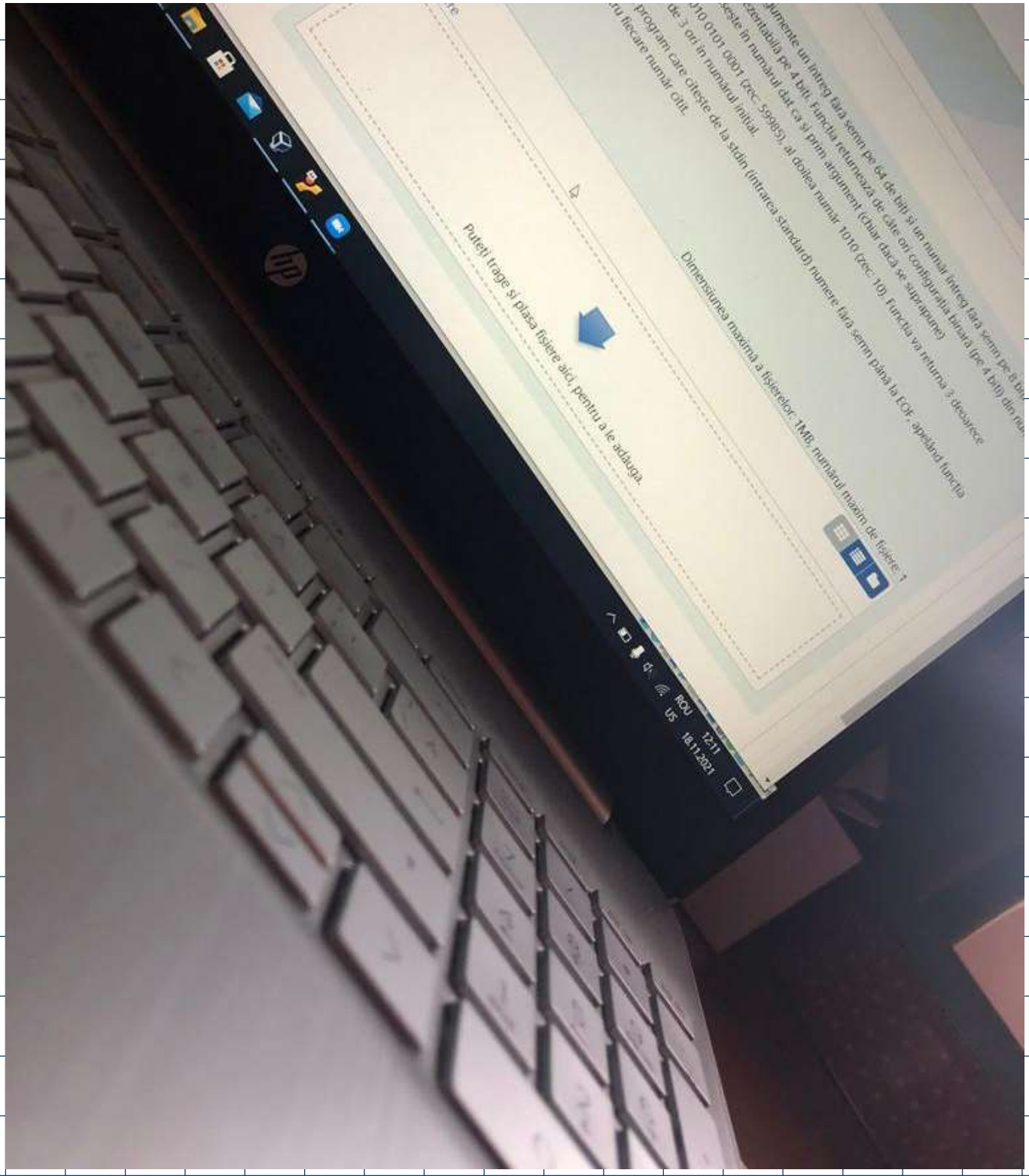
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

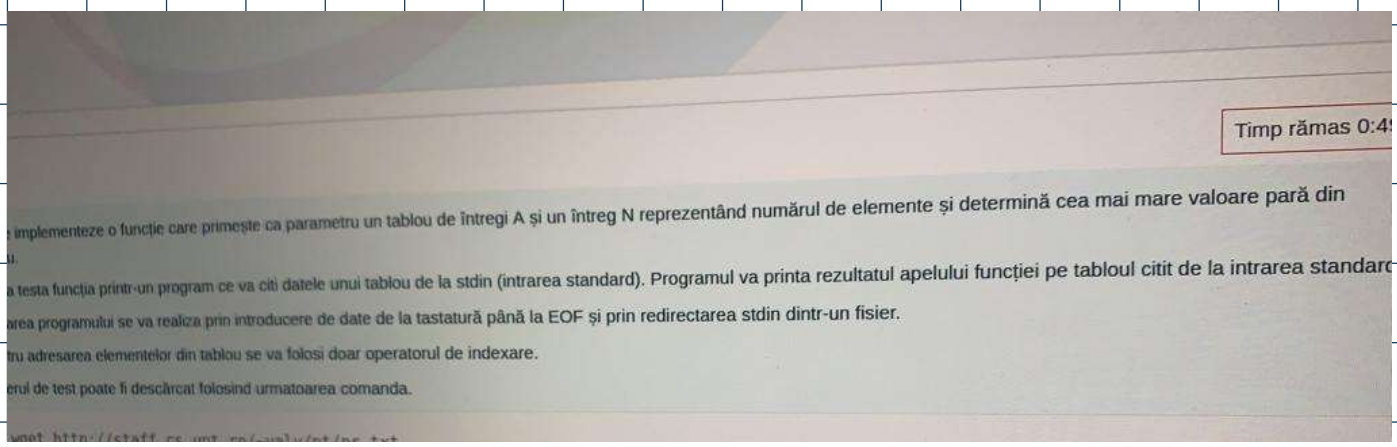


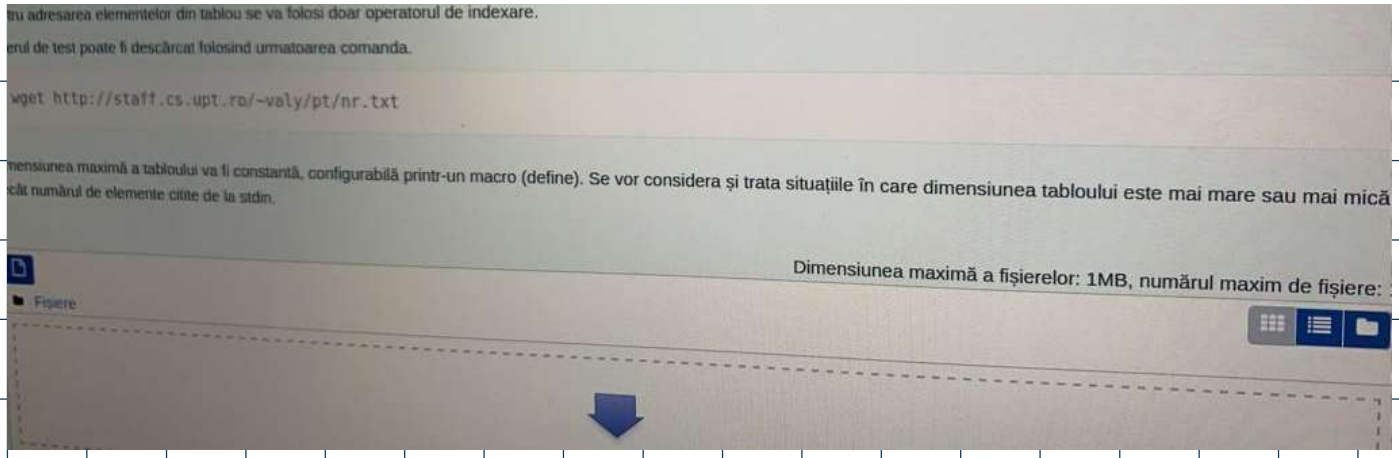

Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
144HZ

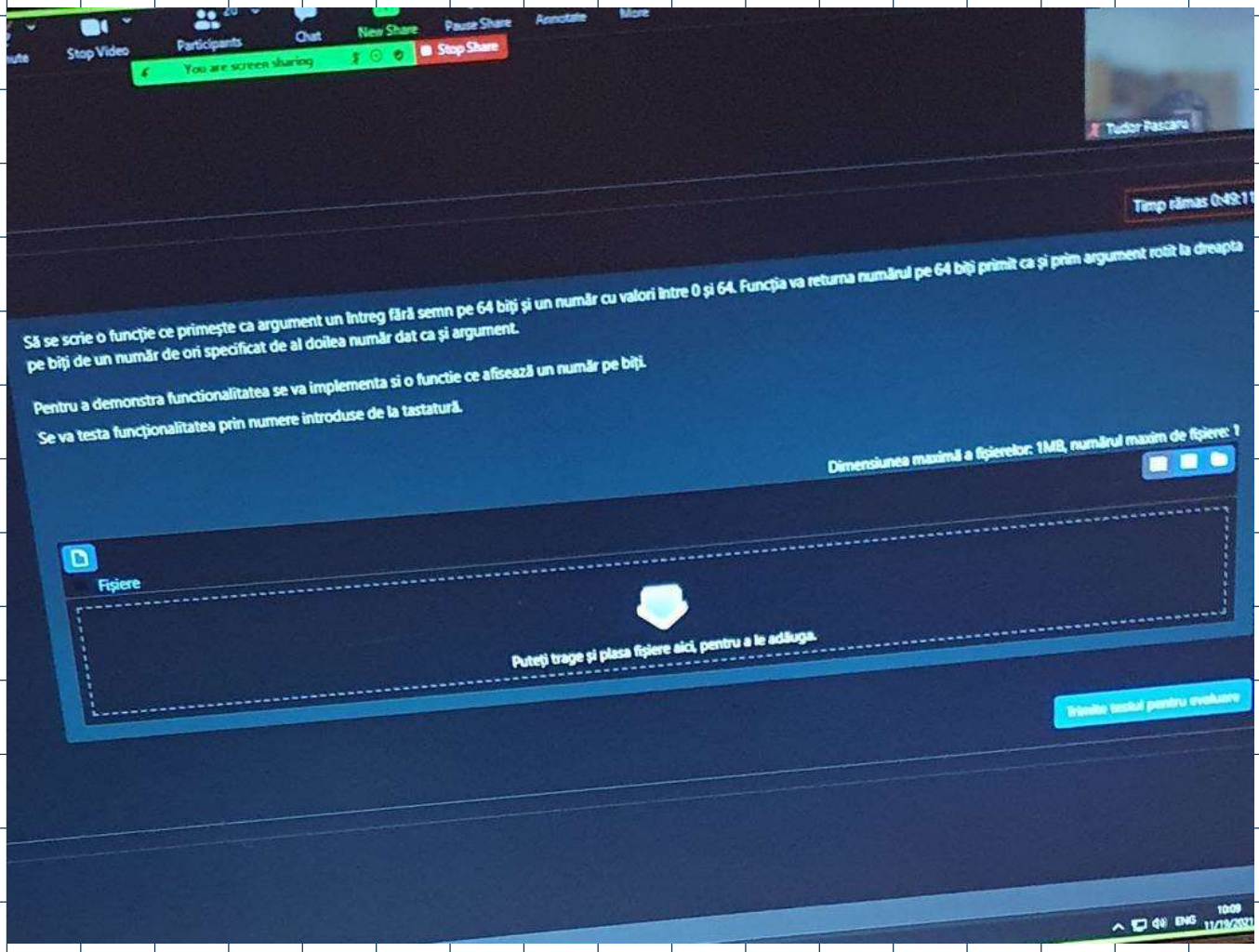

Timp rămas 0:49:41
Se lucrează la un nou sistem de operare numit Polibuntu. Ca în orice sistem de operare, permisiunile fișierelor este un aspect important.
În acest sistem de operare, permisiunile unui fișier sunt stocate într-un număr pe 8 biți fără semn, în 2 nibble-uri de 4 biți: primul nibble reprezintă permisiunile deținătorului (user), iar al doilea nibble reprezintă permisiunile celorlalți useri (other). Cel mai semnificativ bit din nibble nu reprezintă nimic, iar următorii 3 reprezinta în ordine dreptul de read, write, execute. De exemplu:
-rwx -rwx
0111 0100
În exemplul de mai sus, user-ul are drepturi de read, write, execute, iar restul userilor (other), pot doar citi acel fișier.
Să se creeze următoarele funcții:
- o funcție care primește ca parametru un număr întreg pe 8 biți reprezentând permisiunile unui fișier. Funcția va afișa acele permisiuni sub forma -rwx -rwx. De exemplu, pentru numărul 0111 0100, se va afișa: -rwx -r--
- o funcție care primește ca parametru un tablou de astfel de permisiuni, un întreg N reprezentând numărul de elemente, un număr reprezentând permisiunea care va fi adaugată fiecărui număr din tablou fără a modifca restul permisiunilor, și un număr care va determina dacă permisiunea se adauga la nibble-ul user sau la nibble-ul other. De exemplu, dacă adăugăm dreptul de write pentru other pentru numărul 0111 0100, el va deveni 0111 0110.
Se va testa funcționalitatea prin numere introduse de la tastatură.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

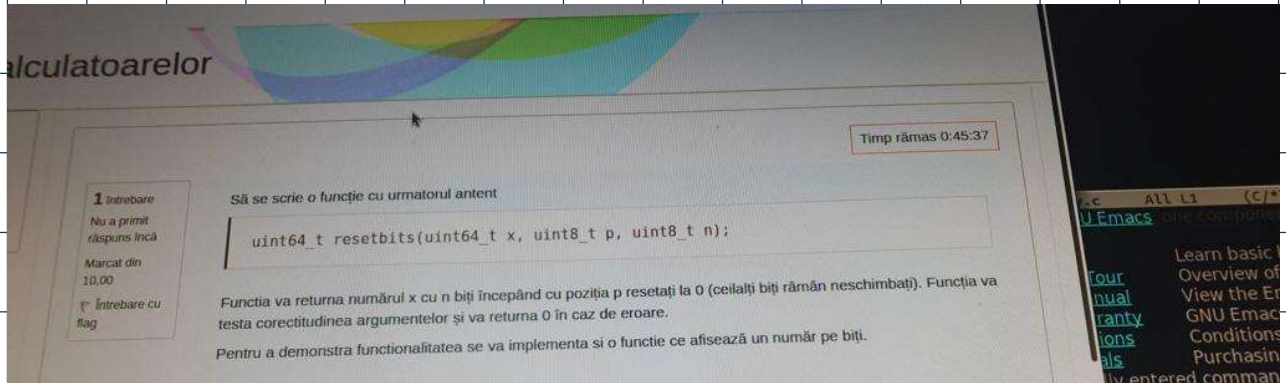

Fișiere

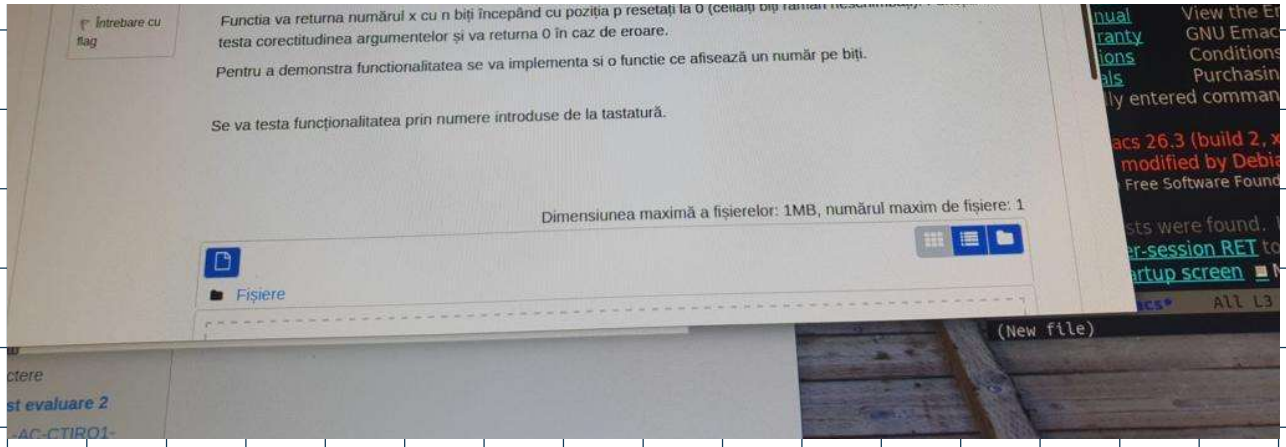

Puteți trage și plasa fișiere aici pentru a le adăuga.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
ROU
US
12:11
18.11.2021

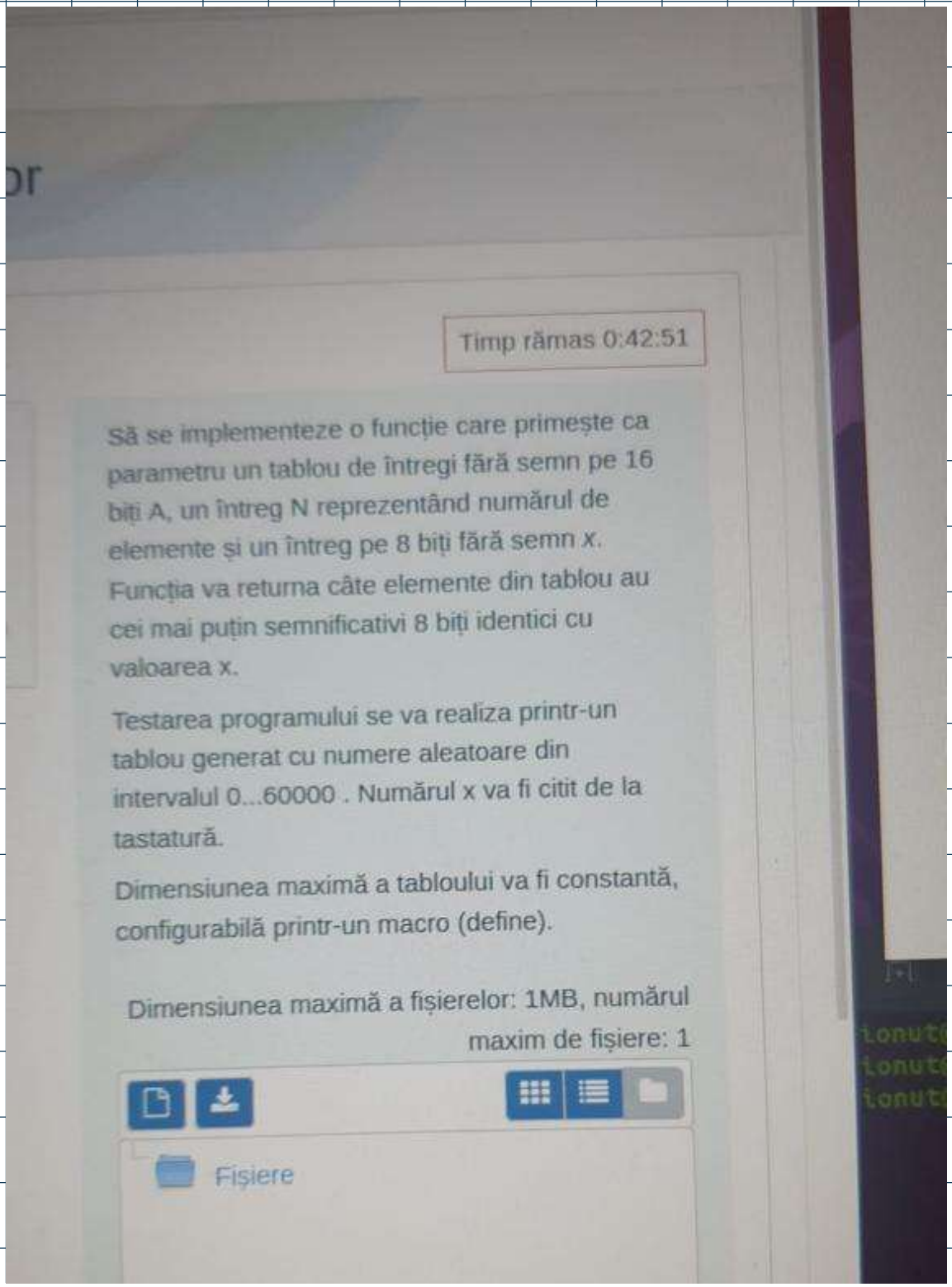

Timp rămas 0:4
implementeze o funcție care primește ca parametru un tablou de întregi A și un întreg N reprezentând numărul de elemente și determină cea mai mare valoare pară din
testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard). Programul va printa rezultatul apelului funcției pe tabloul citit de la intrarea standard
area programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fisier.
tru adresarea elementelor din tablou se va folosi doar operatorul de indexare.
erul de test poate fi descărcat folosind urmatoarea comanda.

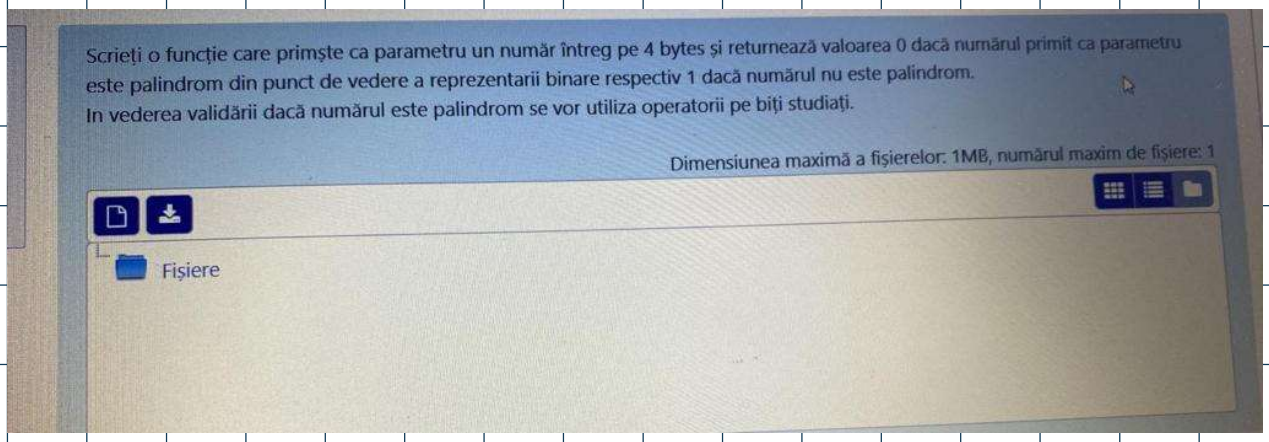

nu adresarea elementelor din tablou se va folosi doar operatorul de indexare.
erul de test poate fi descărcat folosind urmatoarea comanda.
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
mensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică
cât numărul de elemente citite de la stdin.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere:
Fișiere

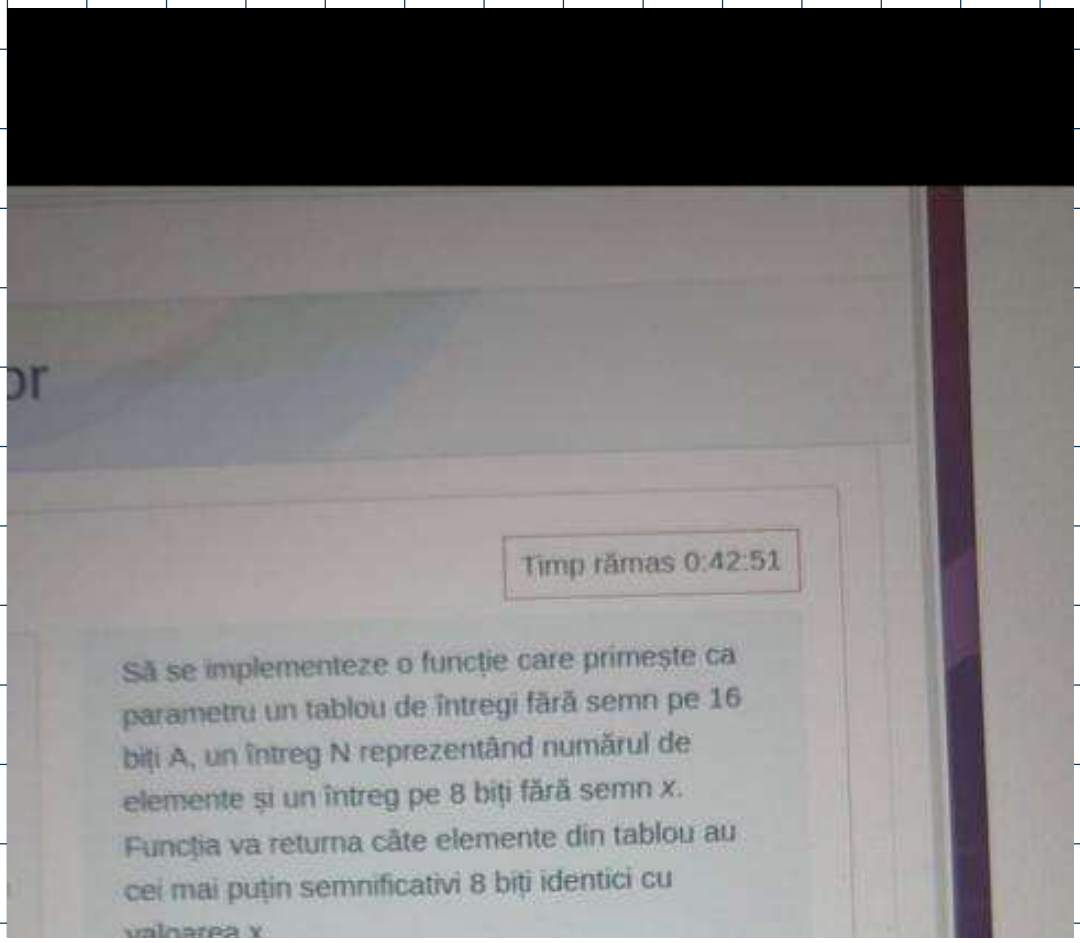

Stop Video
Participants
Chat
New Share
Pause Share
Annotate
More
You are screen sharing
Stop Share
Tudor Pascaru
Timp rămas 0:49:11
Să se scrie o funcție ce primește ca argument un întreg fără semn pe 64 biți și un număr cu valori între 0 și 64. Funcția va returna numărul pe 64 biți primit ca și prim argument rotit la dreapta pe biți de un număr de ori specificat de al doilea număr dat ca și argument.
Pentru a demonstra functionalitatea se va implementa si o functie ce afiseaza un număr pe biți.
Se va testa funcționalitatea prin numere introduse de la tastatură.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Trimite testul pentru evaluare
ENG
10:09

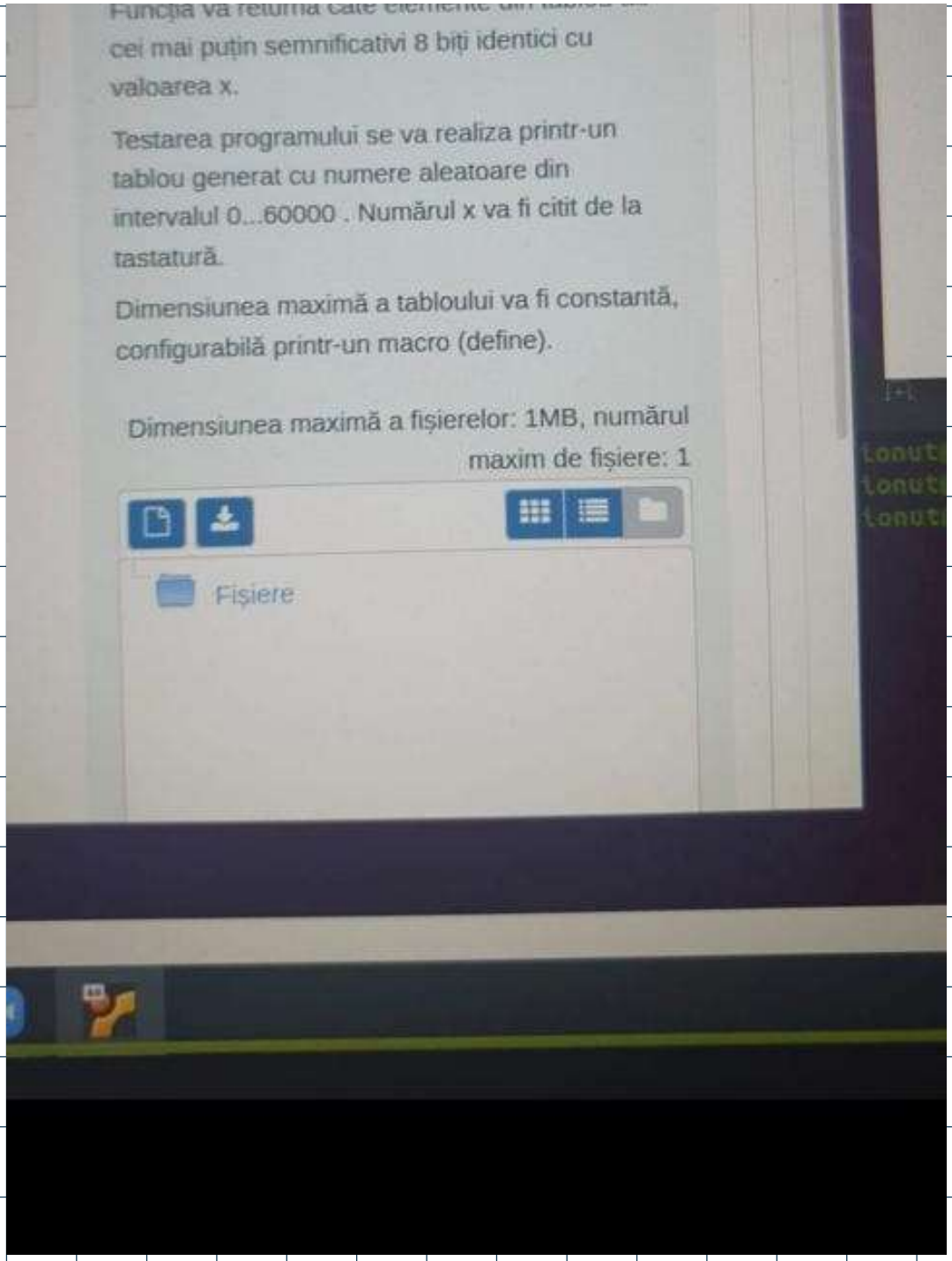

alculatoarelor
Timp rămas 0:45:37
1 Intrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag
Să se scrie o funcție cu urmatorul antent
uint64_t resetbits(uint64_t x, uint8_t p, uint8_t n);
Functia va returna numărul x cu n biți începând cu poziția p resetați la 0 (ceilalți biți rămân neschimbați). Funcția va testa corectitudinea argumentelor și va returna 0 în caz de eroare.
Pentru a demonstra functionalitatea se va implementa si o functie ce afiseaza un număr pe biți.
All L1
U Emacs
Learn basic
Overview of
View the E
GNU Emac
Conditions
Purchasin

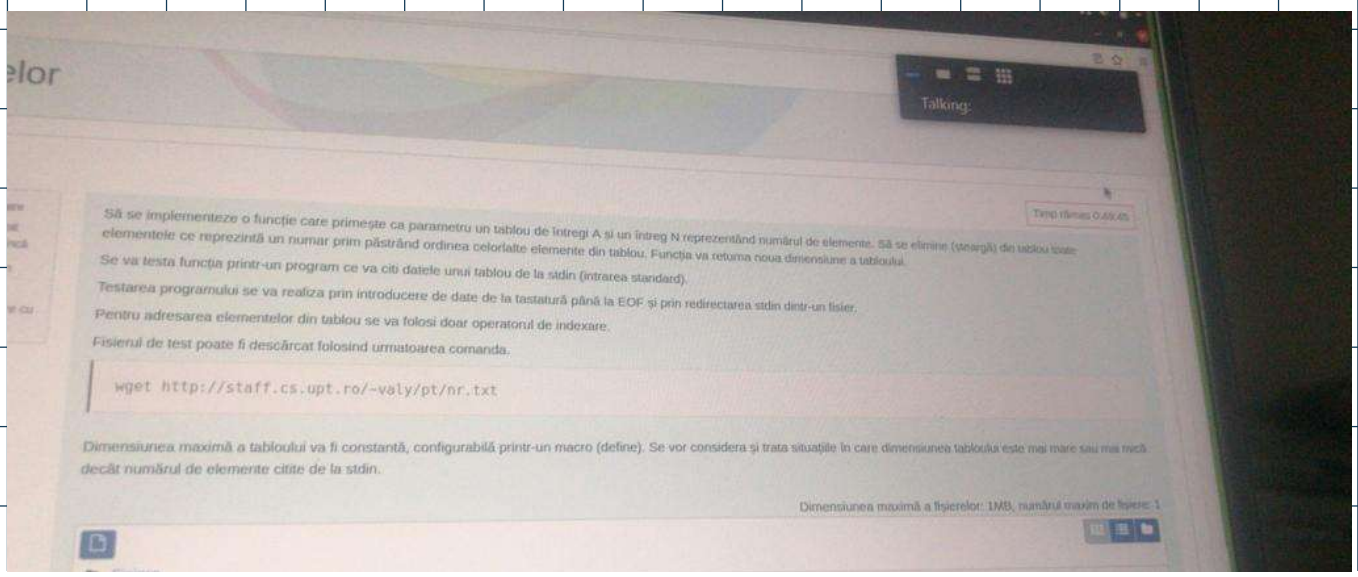

Întrebare cu flag
Functia va returna numărul x cu n biți începând cu poziția p resetați la 0 (ceilalți biți rămân nesch
testa corectitudinea argumentelor și va returna 0 în caz de eroare.
Pentru a demonstra functionalitatea se va implementa si o functie ce afisează un număr pe biți.
Se va testa funcționalitatea prin numere introduse de la tastatură.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
(New file)

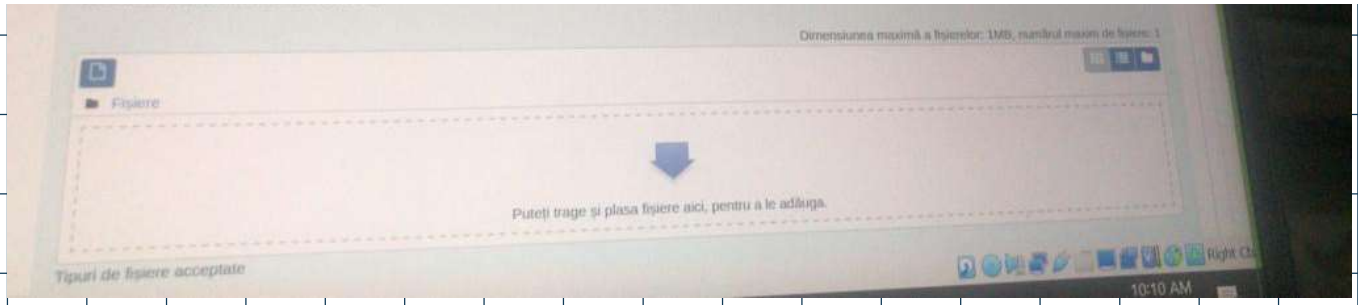

Timp rămas 0:42:51
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 16 biți A, un întreg N reprezentând numărul de elemente și un întreg pe 8 biți fără semn x. Funcția va returna câte elemente din tablou au cei mai puțin semnificativi 8 biți identici cu valoarea x.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...60000 . Numărul x va fi citit de la tastatură.
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere

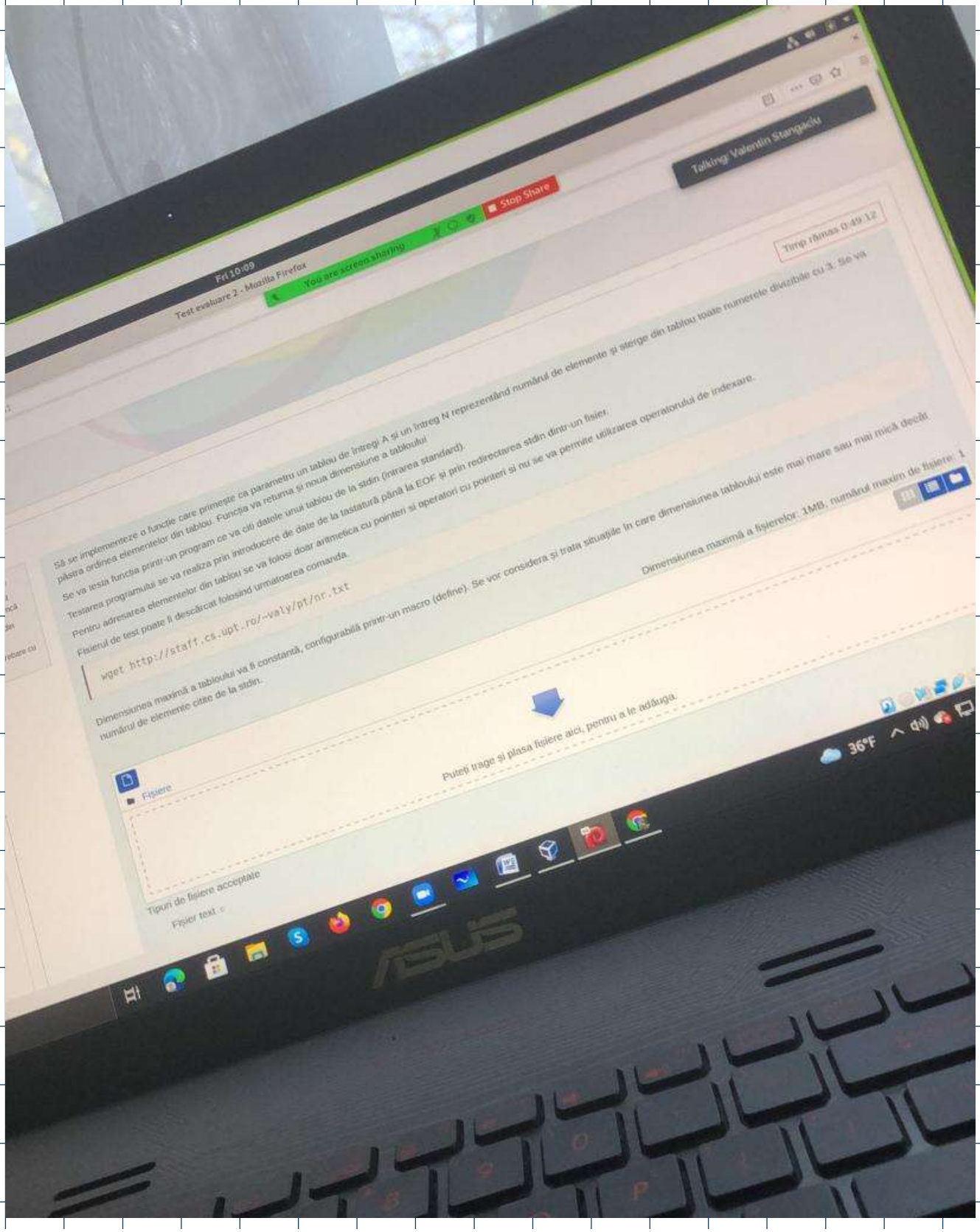

Scrieți o funcție care primște ca parametru un număr întreg pe 4 bytes și returnează valoarea 0 dacă numărul primit ca parametru este palindrom din punct de vedere a reprezentarii binare respectiv 1 dacă numărul nu este palindrom.
In vederea validării dacă numărul este palindrom se vor utiliza operatorii pe biți studiați.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere

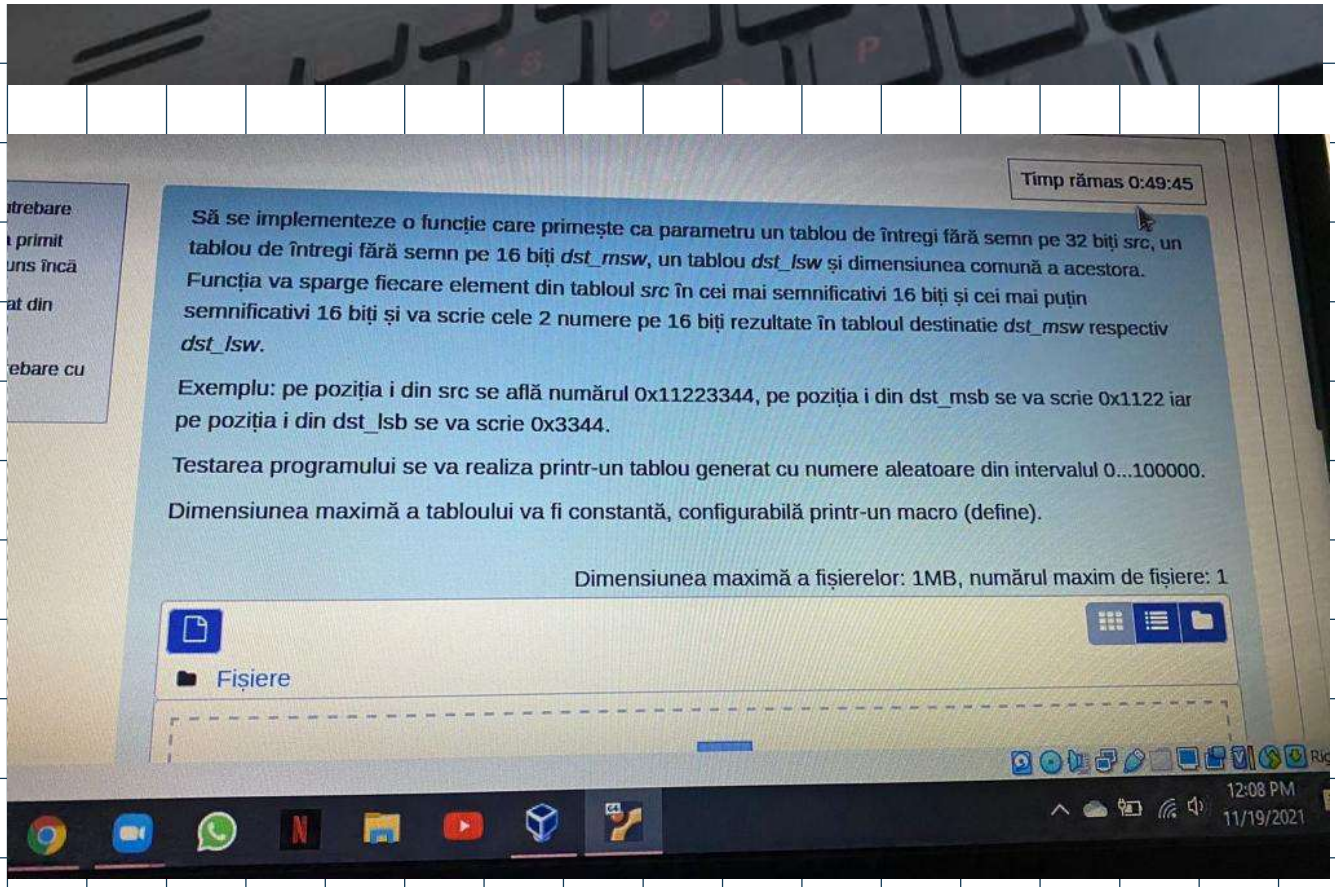

Timp rămas 0:42:51
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 16 biți A, un întreg N reprezentând numărul de elemente și un întreg pe 8 biți fără semn x. Funcția va returna câte elemente din tablou au cei mai puțin semnificativi 8 biți identici cu

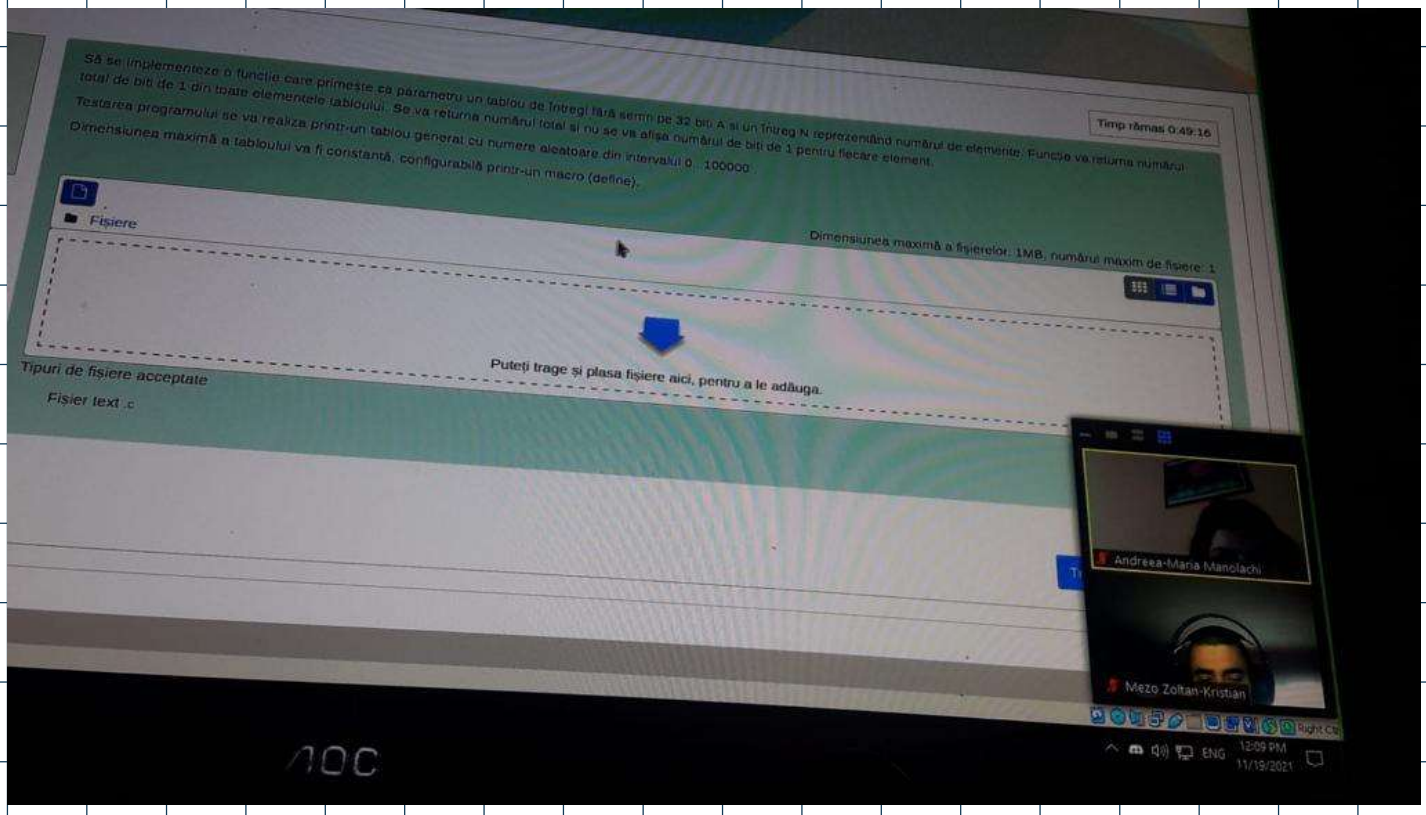

Funcția va returna câte elemente
cei mai puțin semnificativi 8 biți identici cu valoarea x.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...60000 . Numărul x va fi citit de la tastatură.
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere

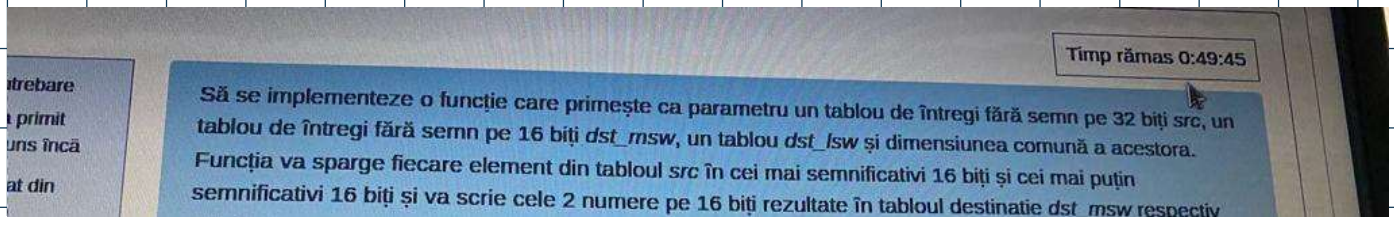

Talking:
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un întreg N reprezentând numărul de elemente. Să se elimine (șteargă) din tablou toate elementele ce reprezintă un număr prim păstrând ordinea celorlalte elemente din tablou. Funcția va returna noua dimensiune a tabloului.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fișier.
Pentru adresarea elementelor din tablou se va folosi doar operatorul de indexare.
Fișierul de test poate fi descărcat folosind următoarea comanda.
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

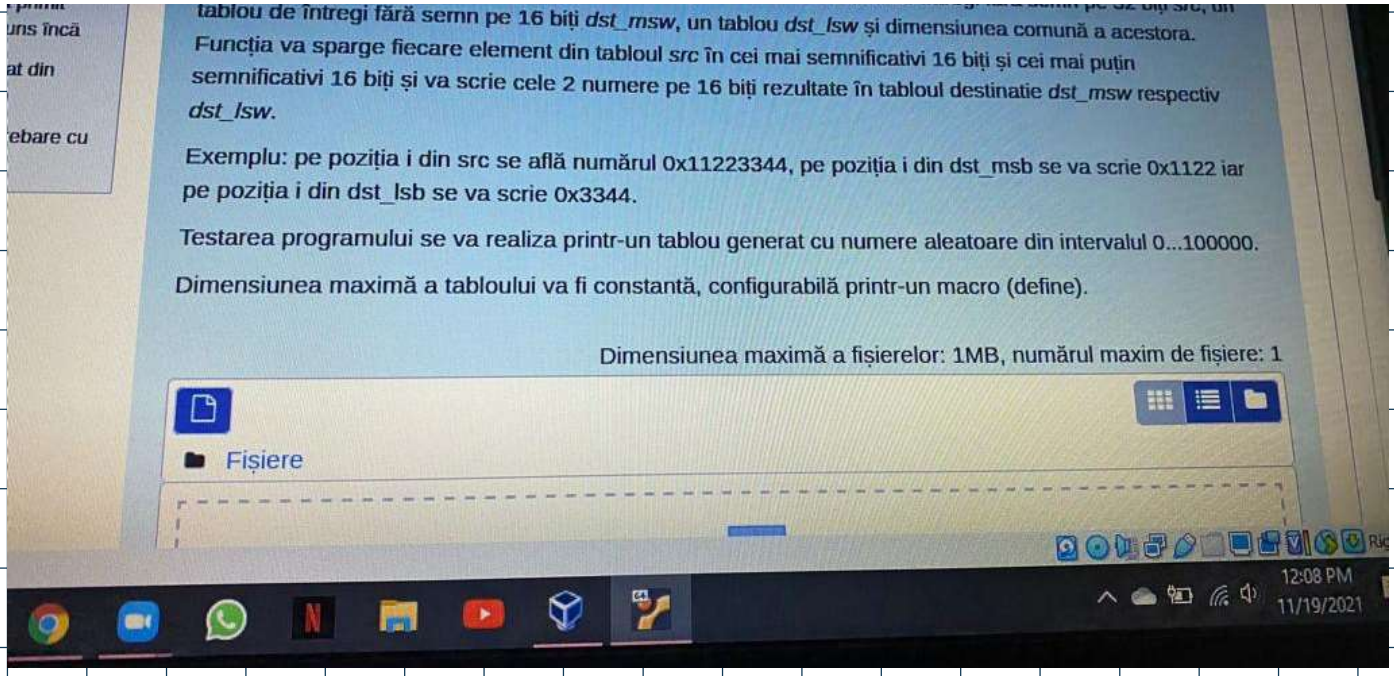

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga
Tipuri de fișiere acceptate
10:10 AM

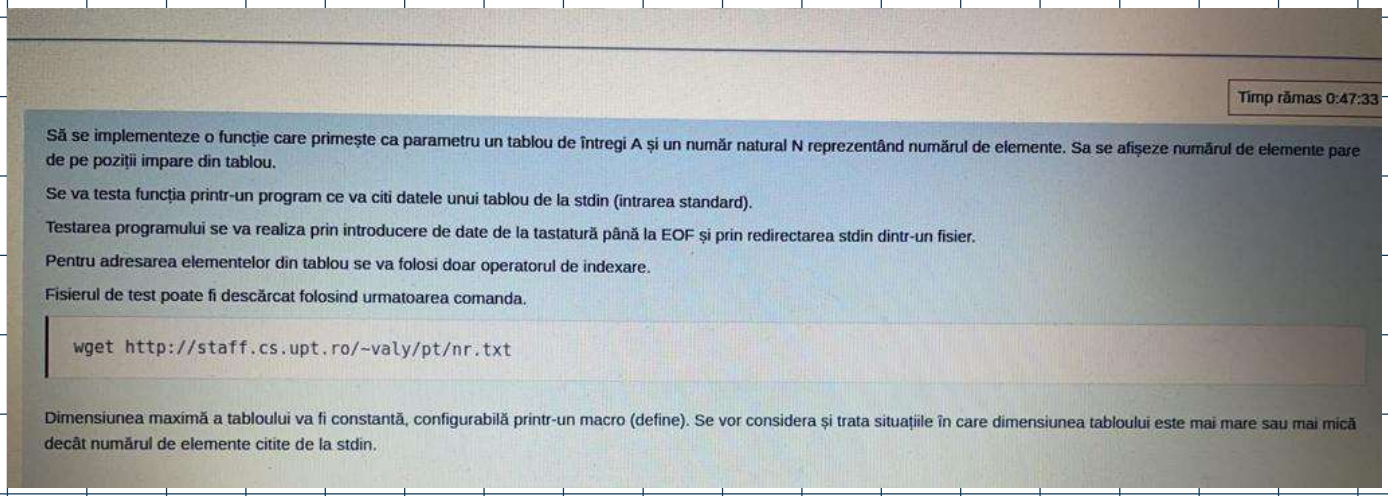

Talking: Valentin Stangaciu
Stop Share
You are screen sharing
Fri 10:09
Test evaluare 2 - Mozilla Firefox
Timp rămas 0:49:12
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un întreg N reprezentând numărul de elemente și șterge din tablou toate numerele divizibile cu 3. Se va păstra ordinea elementelor din tablou. Funcția va returna și noua dimensiune a tabloului.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fișier.
Pentru adresarea elementelor din tablou se va folosi doar aritmetica cu pointeri și operatorii cu pointeri și nu se va permite utilizarea operatorului de indexare.
Fișierul de test poate fi descărcat folosind următoarea comandă
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga
Tipuri de fișiere acceptate
Fișier text
36°F
ASUS

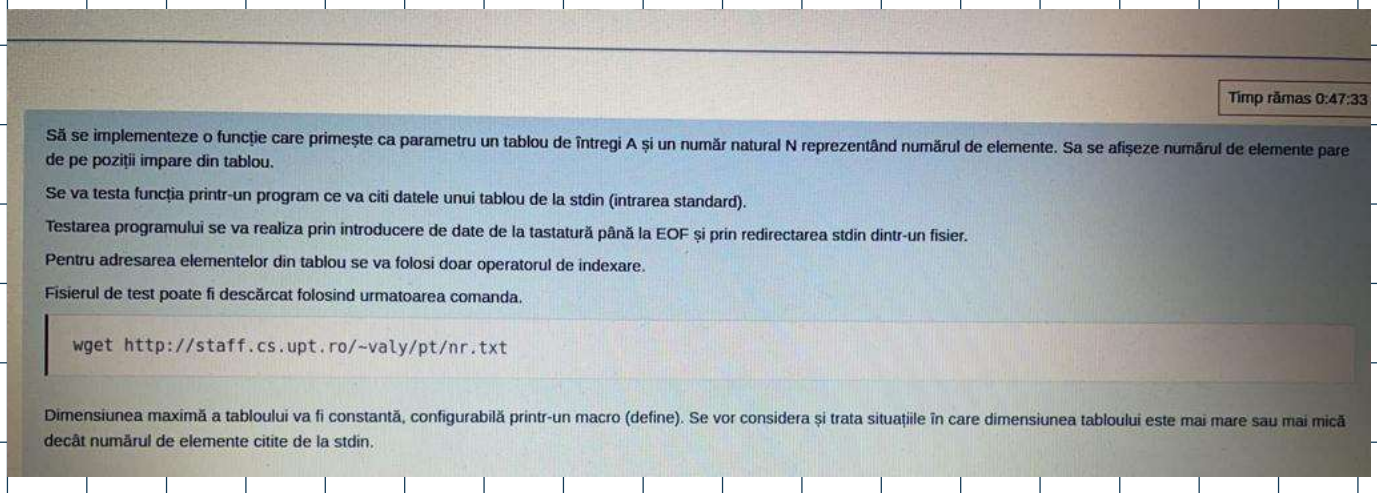

Timp rămas 0:49:45
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 32 biți src, un tablou de întregi fără semn pe 16 biți dst_msw, un tablou dst_lsw și dimensiunea comună a acestora. Funcția va sparge fiecare element din tabloul src în cei mai semnificativi 16 biți și cei mai puțin semnificativi 16 biți și va scrie cele 2 numere pe 16 biți rezultate în tabloul destinatie dst_msw respectiv dst_lsw.
Exemplu: pe poziția i din src se află numărul 0x11223344, pe poziția i din dst_msb se va scrie 0x1122 iar pe poziția i din dst_lsb se va scrie 0x3344.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...100000.
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
12:08 PM
11/19/2021

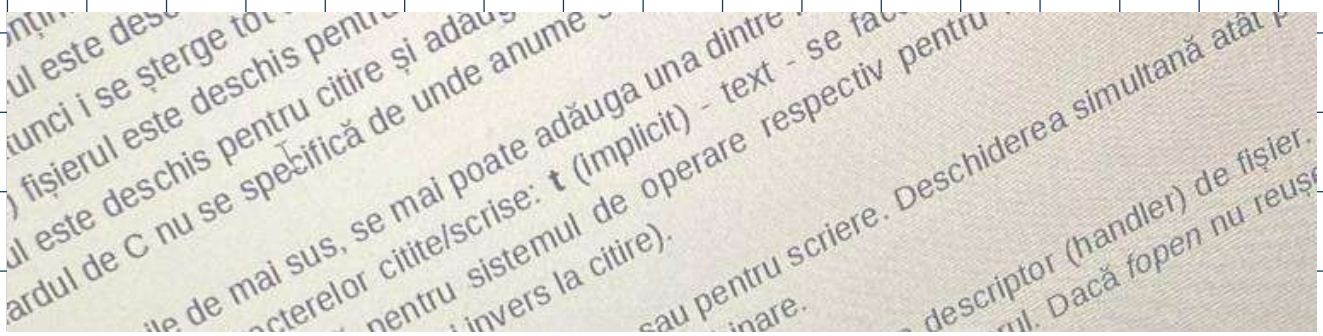

Timp rămas 0:49:16
Fișiere
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Tipuri de fișiere acceptate
Fișier text .c
Andreea-Maria Manolachi
Mezo Zoltan-Kristian
ENG
12:09 PM
11/19/2021

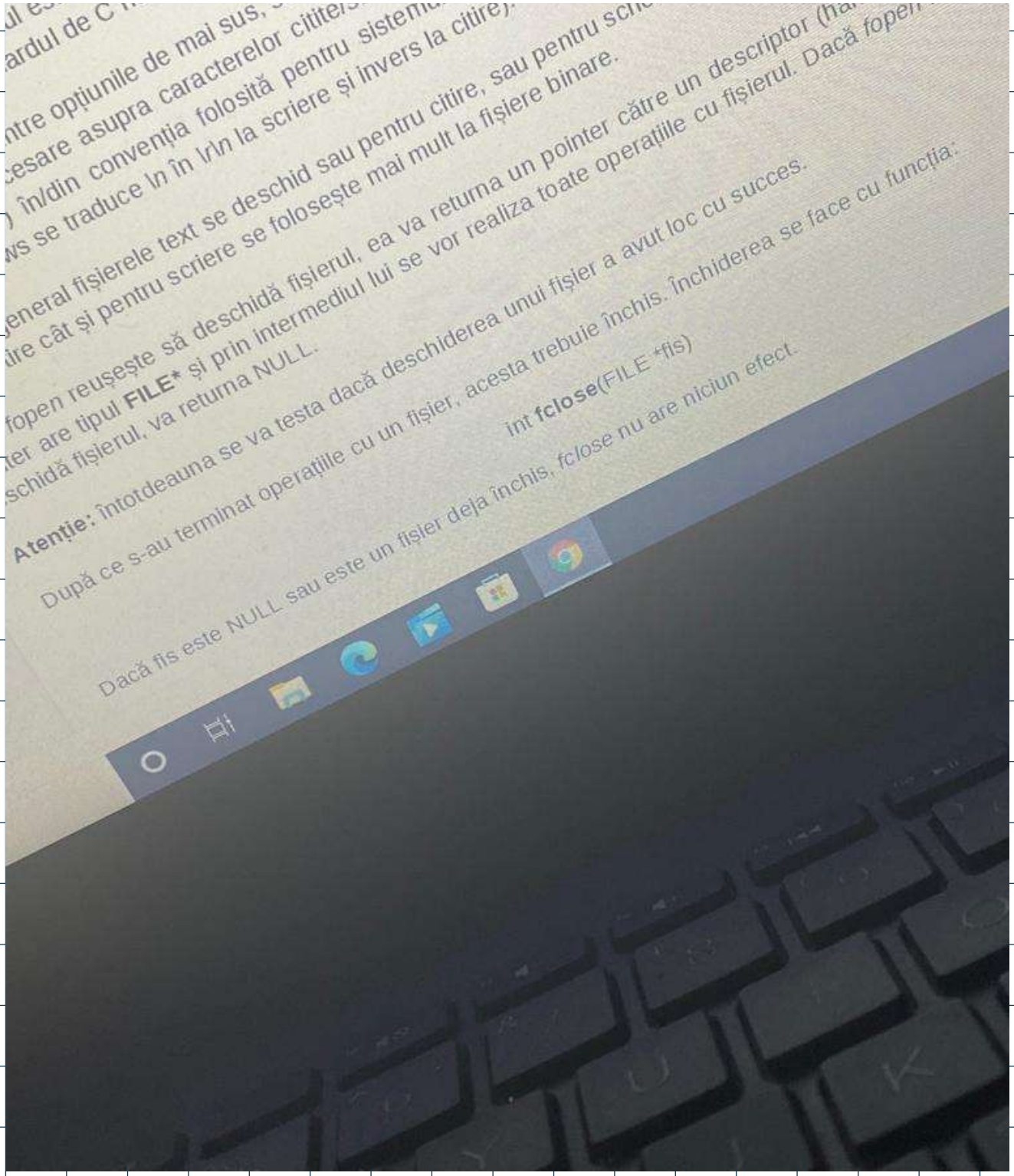

Timp rămas 0:49:45
Să se implementeze o funcție care primește ca parametru un tablou de întregi fără semn pe 32 biți src, un tablou de întregi fără semn pe 16 biți dst_msw, un tablou dst_lsw și dimensiunea comună a acestora.
Funcția va sparge fiecare element din tabloul src în cei mai semnificativi 16 biți și cei mai puțin semnificativi 16 biți și va scrie cele 2 numere pe 16 biți rezultate în tabloul destinatie dst_msw respectiv

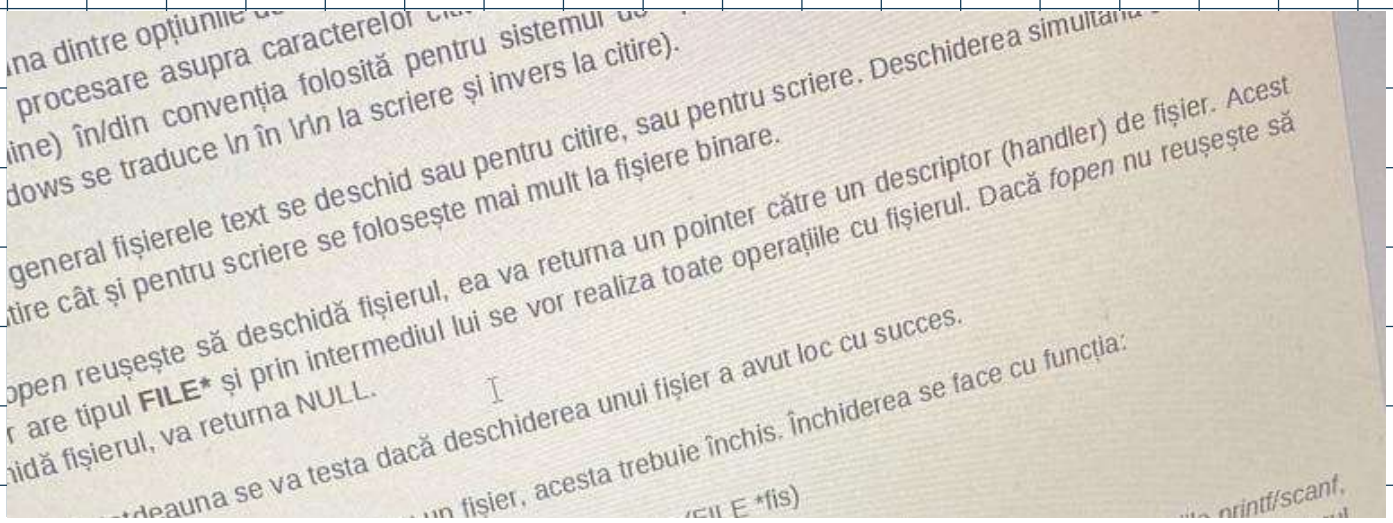

tablou de întregi fără semn pe 16 biți dst_msw, un tablou dst_lsw și dimensiunea comună a acestora.
Funcția va sparge fiecare element din tabloul src în cei mai semnificativi 16 biți și cei mai puțin semnificativi 16 biți și va scrie cele 2 numere pe 16 biți rezultate în tabloul destinatie dst_msw respectiv dst_lsw.
Exemplu: pe poziția i din src se află numărul 0x11223344, pe poziția i din dst_msb se va scrie 0x1122 iar pe poziția i din dst_lsb se va scrie 0x3344.
Testarea programului se va realiza printr-un tablou generat cu numere aleatoare din intervalul 0...100000.
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define).
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
12:08 PM
11/19/2021

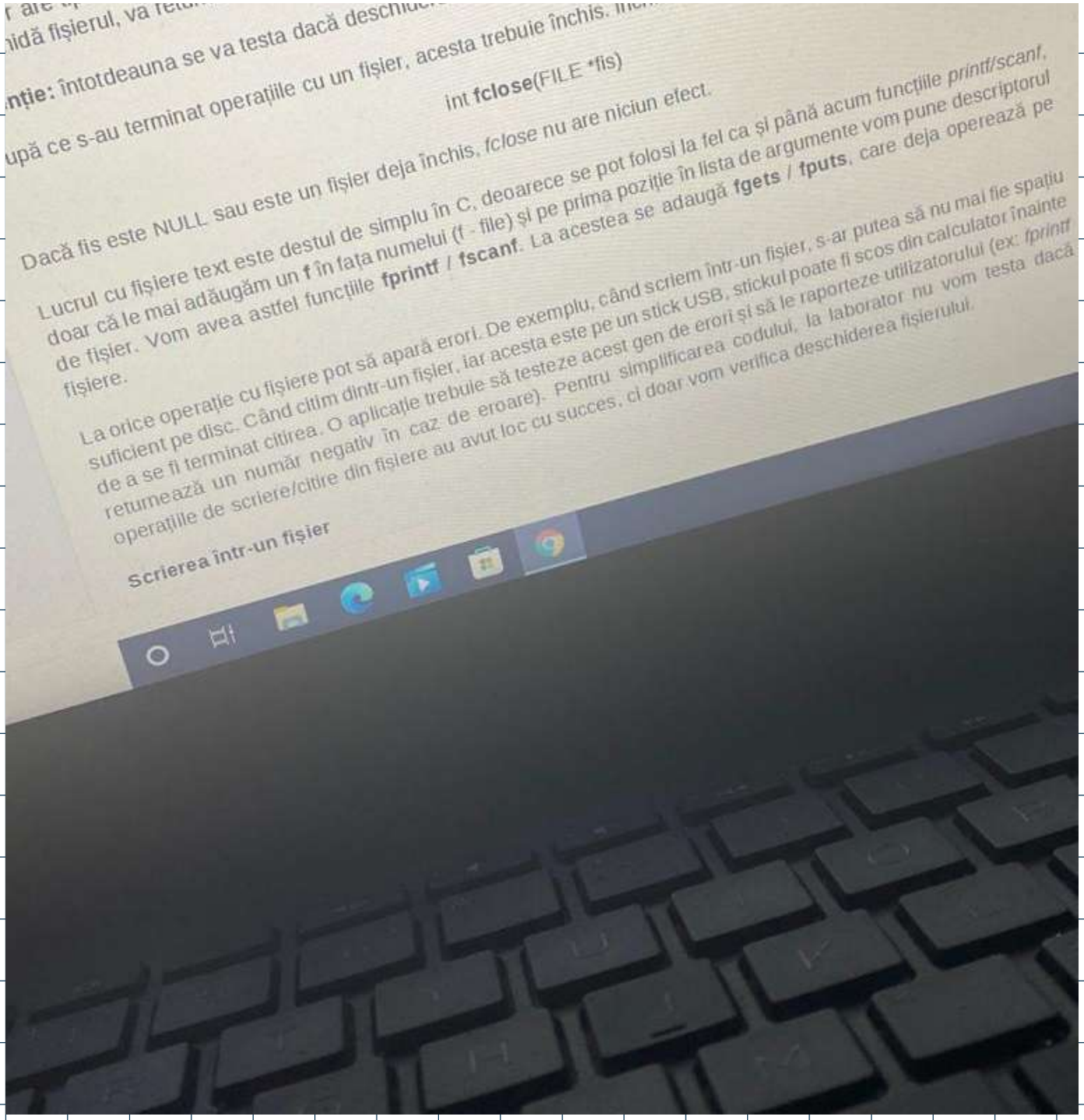

Timp rămas 0:47:33
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un număr natural N reprezentând numărul de elemente. Sa se afișeze numărul de elemente pare de pe poziții impare din tablou.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fisier.
Pentru adresarea elementelor din tablou se va folosi doar operatorul de indexare.
Fisierul de test poate fi descărcat folosind urmatoarea comanda.
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.

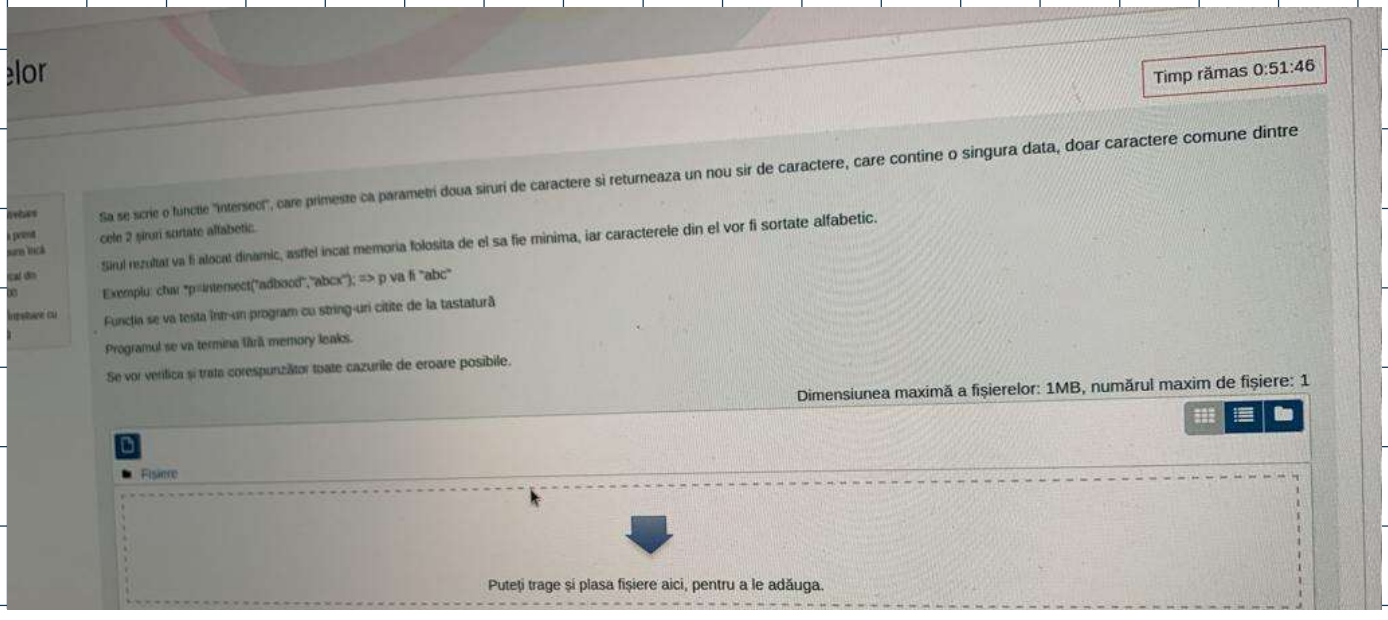

Timp rămas 0:47:33
Să se implementeze o funcție care primește ca parametru un tablou de întregi A și un număr natural N reprezentând numărul de elemente. Sa se afișeze numărul de elemente pare de pe poziții impare din tablou.
Se va testa funcția printr-un program ce va citi datele unui tablou de la stdin (intrarea standard).
Testarea programului se va realiza prin introducere de date de la tastatură până la EOF și prin redirectarea stdin dintr-un fisier.
Pentru adresarea elementelor din tablou se va folosi doar operatorul de indexare.
Fisierul de test poate fi descărcat folosind urmatoarea comanda.
wget http://staff.cs.upt.ro/~valy/pt/nr.txt
Dimensiunea maximă a tabloului va fi constantă, configurabilă printr-un macro (define). Se vor considera și trata situațiile în care dimensiunea tabloului este mai mare sau mai mică decât numărul de elemente citite de la stdin.

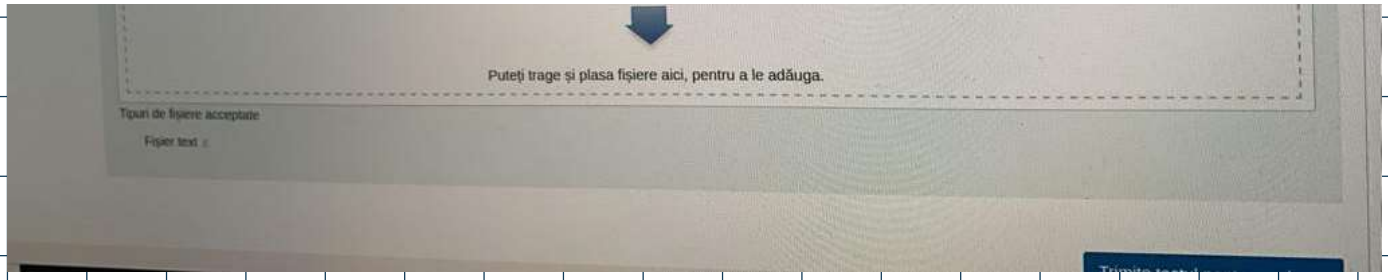


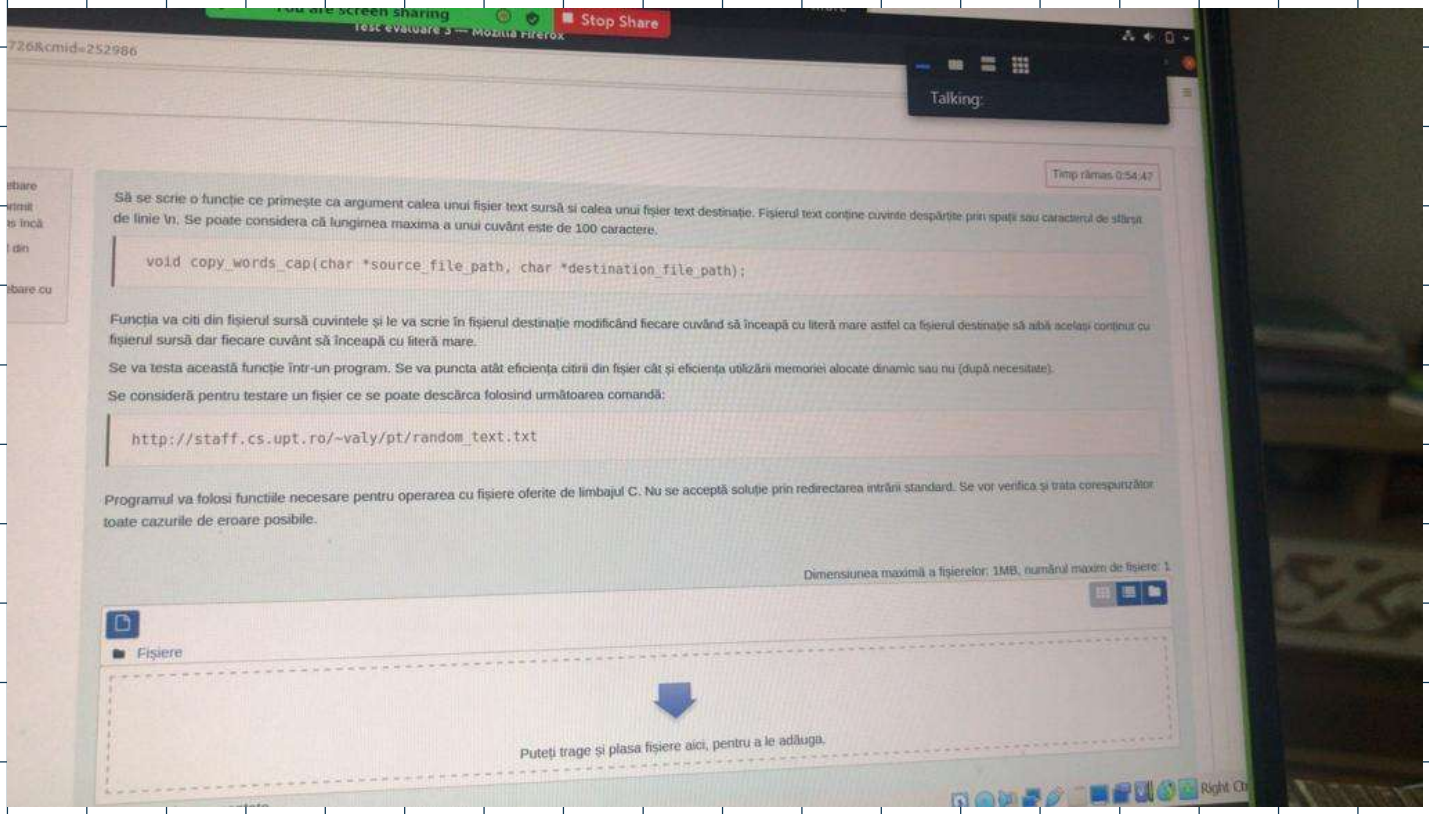

ardul de C
ntre opțiunile de mai sus,
esare asupra caracterelor citite
în/din convenția folosită pentru sistem
ws se traduce \n în \r\n la scriere și invers la citire).
eneral fișierele text se deschid sau pentru citire, sau pentru scr
ire cât și pentru scriere se folosește mai mult la fișiere binare.
fopen reușește să deschidă fișierul, ea va returna un pointer către un descriptor (ha
ter are tipul FILE* și prin intermediul lui se vor realiza toate operațiile cu fișierul. Dacă fopen
schidă fișierul, va returna NULL.
Atenție: întotdeauna se va testa dacă deschiderea unui fișier a avut loc cu succes.
După ce s-au terminat operațiile cu un fișier, acesta trebuie închis. Închiderea se face cu funcția:
int fclose(FILE *fis)
Dacă fis este NULL sau este un fișier deja închis, fclose nu are niciun efect.

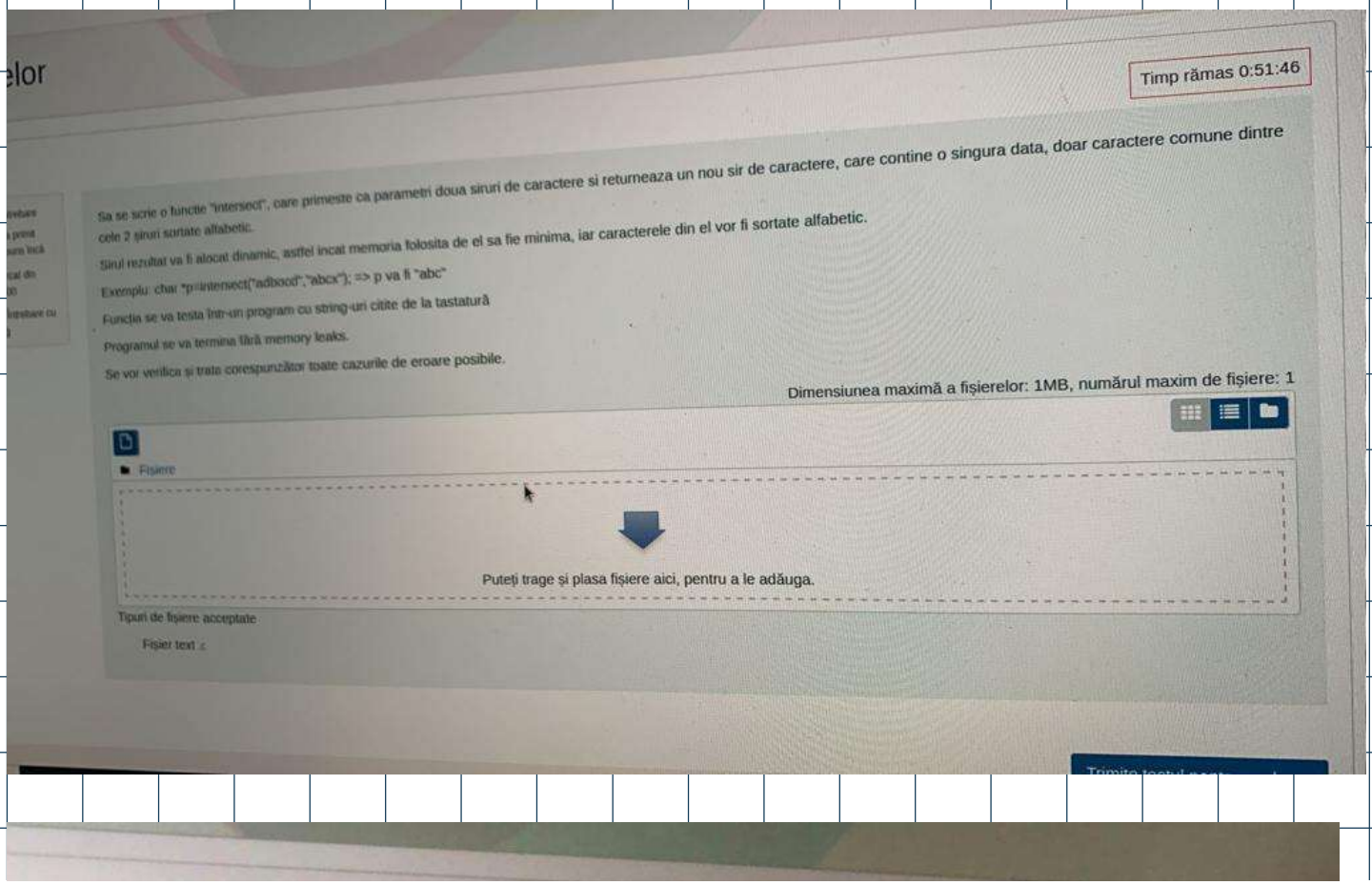

na dintre opțiunile
procesare asupra caracterelor
ine) în/din convenția folosită pentru sistemul
dows se traduce \n în \r\n la scriere și invers la citire).
general fișierele text se deschid sau pentru citire, sau pentru scriere. Deschiderea simultană
tire cât și pentru scriere se folosește mai mult la fișiere binare.
pen reușește să deschidă fișierul, ea va returna un pointer către un descriptor (handler) de fișier. Acest
r are tipul FILE* și prin intermediul lui se vor realiza toate operațiile cu fișierul. Dacă fopen nu reușește să
hidă fișierul, va returna NULL.
deauna se va testa dacă deschiderea unui fișier a avut loc cu succes.
un fișier, acesta trebuie închis. Închiderea se face cu funcția:
FILE *fis)
printf/scanf,

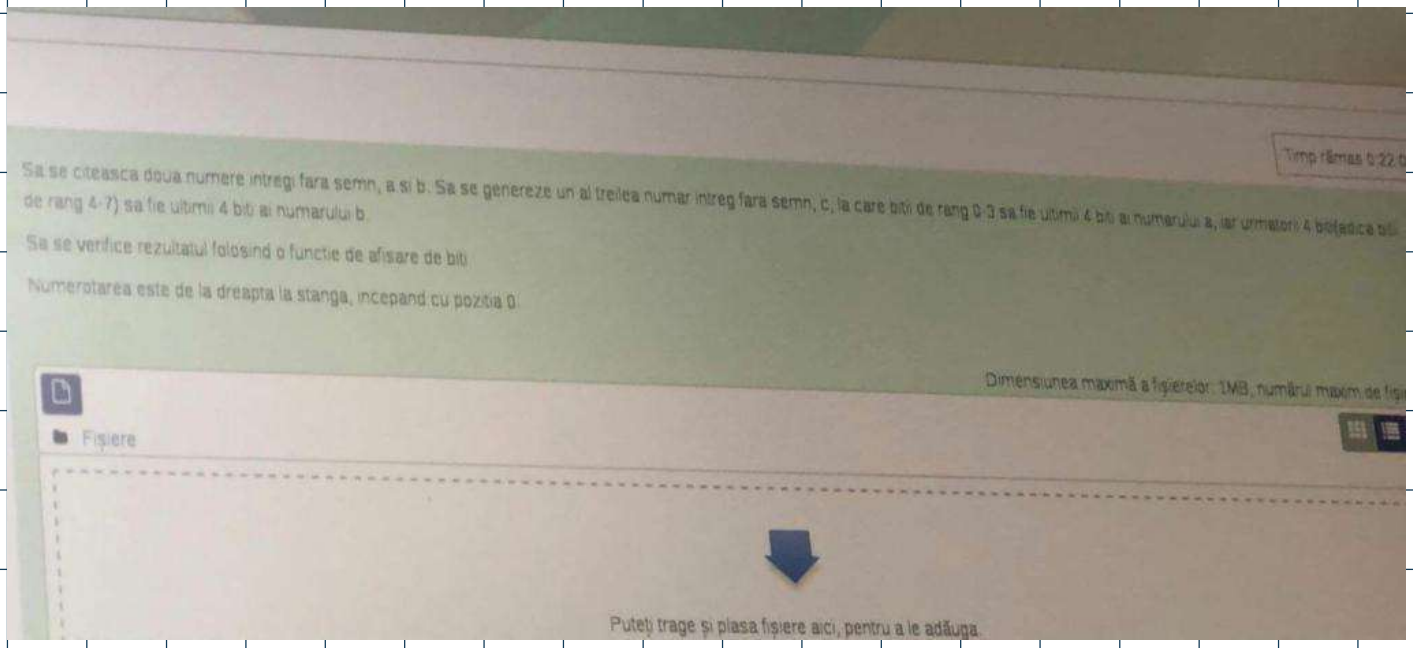

nță: întotdeauna se va testa dacă deschide
upă ce s-au terminat operațiile cu un fișier, acesta trebuie închis. Înch
int fclose(FILE *fis)
Dacă fis este NULL sau este un fișier deja închis, fclose nu are niciun efect.
Lucrul cu fișiere text este destul de simplu în C, deoarece se pot folosi la fel ca și până acum funcțiile printf/scanf, doar că le mai adăugăm un f în fața numelui (f - file) și pe prima poziție în lista de argumente vom pune descriptorul de fișier. Vom avea astfel funcțiile fprintf / fscanf. La acestea se adaugă fgets / fputs, care deja operează pe fișiere.
La orice operație cu fișiere pot să apară erori. De exemplu, când scriem într-un fișier, s-ar putea să nu mai fie spațiu suficient pe disc. Când citim dintr-un fișier, iar acesta este pe un stick USB, stickul poate fi scos din calculator înainte de a se fi terminat citirea. O aplicație trebuie să testeze acest gen de erori și să le raporteze utilizatorului (ex: fprintf returnează un număr negativ în caz de eroare). Pentru simplificarea codului, la laborator nu vom testa dacă operațiile de scriere/citire din fișiere au avut loc cu succes, ci doar vom verifica deschiderea fișierului.
Scrierea într-un fișier

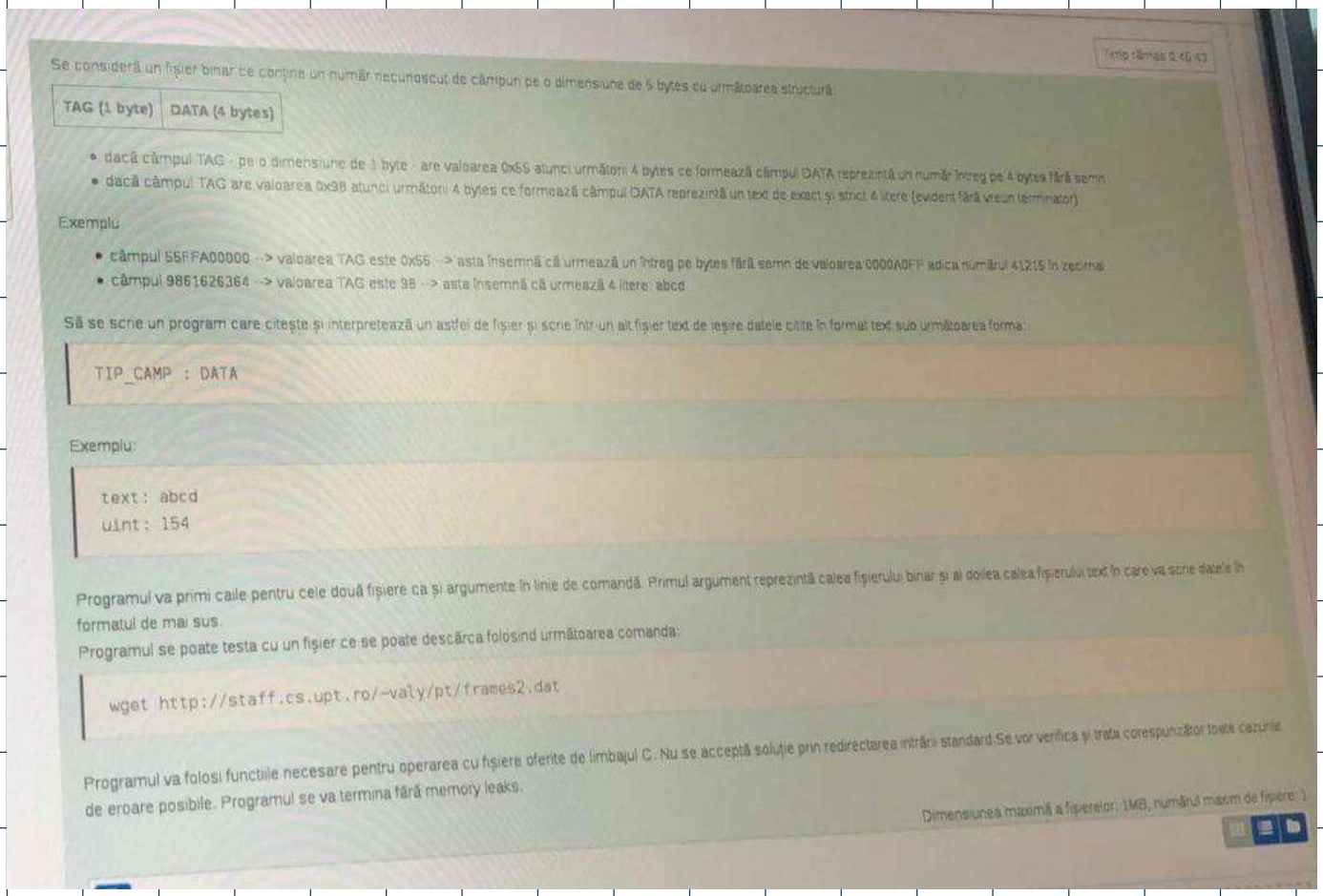

elor
Timp rămas 0:51:46
Sa se scrie o functie "intersect", care primeste ca parametri doua siruri de caractere si returneaza un nou sir de caractere, care contine o singura data, doar caractere comune dintre cele 2 șiruri sortate alfabetic.
Sirul rezultat va fi alocat dinamic, astfel incat memoria folosita de el sa fie minima, iar caracterele din el vor fi sortate alfabetic.
Exemplu: char *p=intersect("adbxxd","abcx"); => p va fi "abc"
Funcția se va testa într-un program cu string-uri citite de la tastatură
Programul se va termina fără memory leaks.
Se vor verifica și trata corespunzător toate cazurile de eroare posibile.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.

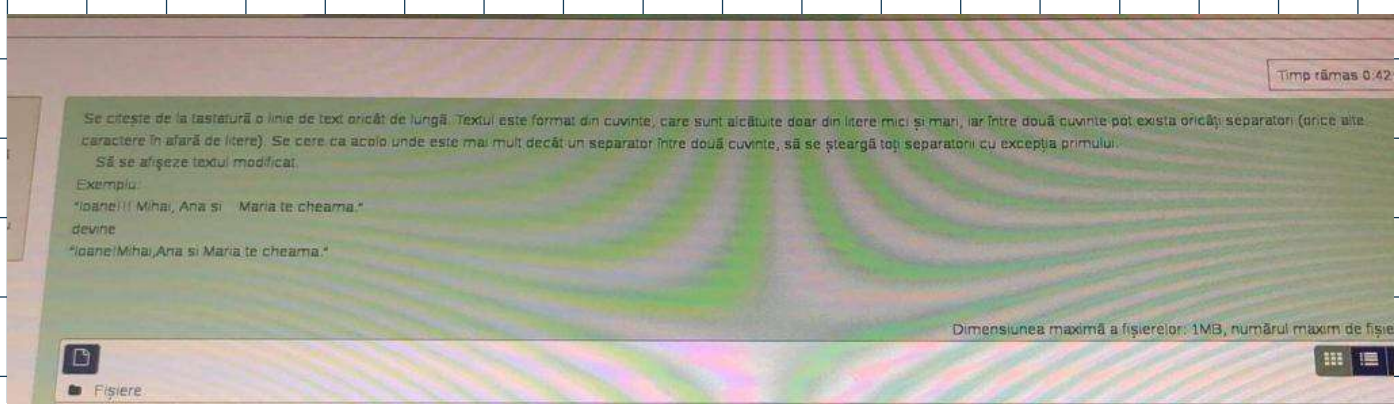

Puteți trage și plasa fișiere aici, pentru a le adăuga.
Tipuri de fișiere acceptate
Fișier text

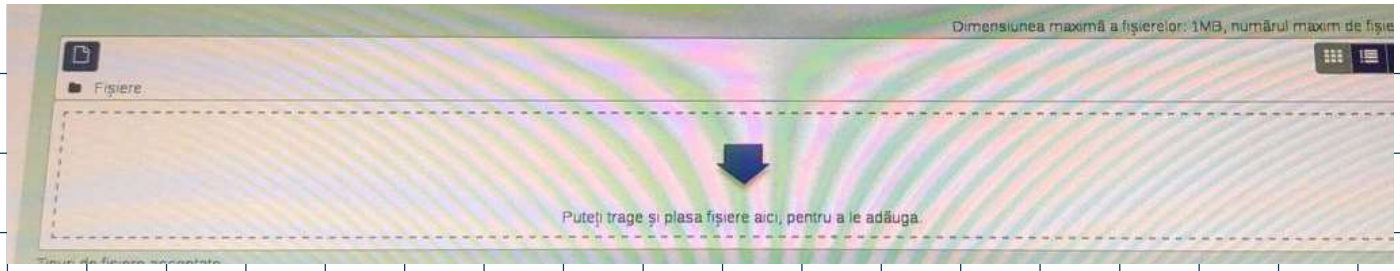

Stop Share
Talking:
Timp rămas 0:54:47
Să se scrie o funcție ce primește ca argument calea unui fișier text sursă si calea unui fișier text destinație. Fișierul text conține cuvinte despărțite prin spații sau caracterul de sfârșit de linie \n. Se poate considera că lungimea maxima a unui cuvânt este de 100 caractere.
void copy_words_cap(char *source_file_path, char *destination_file_path);
Funcția va citi din fișierul sursă cuvintele și le va scrie în fișierul destinație modificând fiecare cuvând să înceapă cu literă mare astfel ca fișierul destinație să aibă același conținut cu fișierul sursă dar fiecare cuvânt să înceapă cu literă mare.
Se va testa această funcție într-un program. Se va puncta atât eficiența citirii din fișier cât și eficiența utilizării memoriei alocate dinamic sau nu (după necesitate).
Se consideră pentru testare un fișier ce se poate descărca folosind următoarea comandă:
http://staff.cs.upt.ro/~valy/pt/random_text.txt
Programul va folosi functiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard. Se vor verifica și trata corespunzător toate cazurile de eroare posibile.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.


Timp rămas 0:51:46
Sa se scrie o functie "intersect", care primeste ca parametri doua siruri de caractere si returneaza un nou sir de caractere, care contine o singura data, doar caractere comune dintre cele 2 șiruri sortate alfabetic.
Sirul rezultat va fi alocat dinamic, astfel incat memoria folosita de el sa fie minima, iar caracterele din el vor fi sortate alfabetic.
Exemplu: char *p=intersect("adbood","abcx"); => p va fi "abc"
Funcția se va testa într-un program cu string-uri citite de la tastatură
Programul se va termina fără memory leaks.
Se vor verifica si trata corespunzător toate cazurile de eroare posibile.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.
Tipuri de fișiere acceptate
Fișier text


Timp rămas 0:22
Să se citească două numere întregi fără semn, a și b. Să se genereze un al treilea număr întreg fără semn, c, la care biții de rang 0-3 să fie ultimii 4 biți ai numărului a, iar următorii 4 biți (adică biți
de rang 4-7) să fie ultimii 4 biți ai numărului b.
Să se verifice rezultatul folosind o funcție de afișare de biți.
Numerotarea este de la dreapta la stânga, începând cu poziția 0.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișie
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga.


Timp rămas 0:46:43
Se consideră un fișier binar ce conține un număr necunoscut de câmpuri pe o dimensiune de 5 bytes cu următoarea structură:
TAG (1 byte)
DATA (4 bytes)
dacă câmpul TAG - pe o dimensiune de 1 byte - are valoarea 0x65 atunci următorii 4 bytes ce formează câmpul DATA reprezintă un număr întreg pe 4 bytes fără semn
dacă câmpul TAG are valoarea 0x98 atunci următorii 4 bytes ce formează câmpul DATA reprezintă un text de exact și strict 4 litere (evident fără vreun terminator)
Exemplu
câmpul 55FFA00000 --> valoarea TAG este 0x55 --> asta înseamnă că urmează un întreg pe bytes fără semn de valoarea 0000A0FF adica numărul 41215 în zecimal
câmpul 9861626364 --> valoarea TAG este 98 --> asta înseamnă că urmează 4 litere: abcd
Să se scrie un program care citește și interpretează un astfel de fișier și scrie într-un alt fișier text de ieșire datele citite în format text sub următoarea forma:
TIP_CAMP : DATA
Exemplu:
text: abcd
uint: 154
Programul va primi caile pentru cele două fișiere ca și argumente în linie de comandă. Primul argument reprezintă calea fișierului binar și al doilea calea fișierului text în care va scrie datele în formatul de mai sus.
Programul se poate testa cu un fișier ce se poate descărca folosind următoarea comanda:
wget http://staff.cs.upt.ro/~valy/pt/frames2.dat
Programul va folosi funcțiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard.Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks.
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1


Timp rămas 0:42
Se citește de la tastatură o linie de text oricât de lungă. Textul este format din cuvinte, care sunt alcătuite doar din litere mici și mari, iar între două cuvinte pot exista oricâți separatori (orice alte caractere în afară de litere). Se cere ca acolo unde este mai mult decât un separator între două cuvinte, să se șteargă toți separatorii cu excepția primului.
Să se afișeze textul modificat.
Exemplu:
"Ioane!!! Mihai, Ana si Maria te cheama."
devine
"Ioane!Mihai,Ana si Maria te cheama."
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișie
Fișiere



Timp rămas 1:02:40

Se consideră un fișier care conține pe câte o linie un nume, format dintr-un singur cuvânt scris doar cu majuscule urmat de un număr de identificare (ID) despărțite prin spațiu. Nu se cunoaște dimensiunea fișierului (numărul de linii).

Exemplu:

```
ION 876
ANDREEA 345
```

Să se proiecteze un tip de date utilizator capabil să implementeze și să stocheze o înregistrare (o linie) cât mai abstract. Se consideră că numele nu poate depăși 50 de caractere. Să se scrie un program care citește un asemenea fișier a cărui cale este dată ca fiind primul argument în linie de comandă. Programul va stoca înregistrările din fișier într-un tablou alocat dinamic, de tipul proiectat.

Programul va mai primi încă două argumente în linie de comandă. Argumentul 2 este reprezentat de calea unui fișier de ieșire. Argumentul 3 este reprezentat de un număr pozitiv sau negativ. Daca argumentul 3 este un număr pozitiv (0 este considerat număr pozitiv) atunci programul va scrie în fișierul de ieșire tabloul cu date ordonat crescător după nume alfabetic, pe câte o linie fiecare înregistrare sub forma:

```
NUME : ID
```

Dacă argumentul 3 este un număr negativ atunci programul va scrie în fișierul de ieșire tabloul cu date ordonat crescător după ID sub forma:

```
ID : NUME
```

Pentru testarea programului se poate folosi un fișier ce se poate descărca folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/name_list.txt
```

Programul va folosi functiile necesare pentru operarea cu fișiere text oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard. Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks.Se notează și elemente de eficiență. Se consideră ca fiind necunoscută dimensiunea fișierului de intrare. Pentru sortare se poate adapta funcția Bubble Sort disponibilă în secțiunea Resurse Laborator.

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

Timp rămas 1:02:40

Se consideră un fișier care conține pe câte o linie un nume, format dintr-un singur cuvânt scris doar cu majuscule urmat de un număr de identificare (ID) despărțite prin spațiu. Nu se cunoaște dimensiunea fișierului (numărul de linii).

Exemplu:

```
ION 876
ANDREEA 345
```

Să se proiecteze un tip de date utilizator capabil să implementeze și să stocheze o înregistrare (o linie) cât mai abstract. Se consideră că numele nu poate depăși 50 de caractere. Să se scrie un program care citește un asemenea fișier a cărui cale este dată ca fiind primul argument în linie de comandă. Programul va stoca înregistrările din fișier într-un tablou alocat dinamic, de tipul proiectat.

Programul va mai primi încă două argumente în linie de comandă. Argumentul 2 este reprezentat de calea unui fișier de ieșire. Argumentul 3 este reprezentat de un număr pozitiv sau negativ. Daca argumentul 3 este un număr pozitiv (0 este considerat număr pozitiv) atunci programul va scrie în fișierul de ieșire tabloul cu date ordonat crescător după nume alfabetic, pe câte o linie fiecare înregistrare sub forma:

```
NUME : ID
```

Dacă argumentul 3 este un număr negativ atunci programul va scrie în fișierul de ieșire tabloul cu date ordonat crescător după ID sub forma:

```
ID : NUME
```

Pentru testarea programului se poate folosi un fișier ce se poate descărca folosind următoarea comandă:

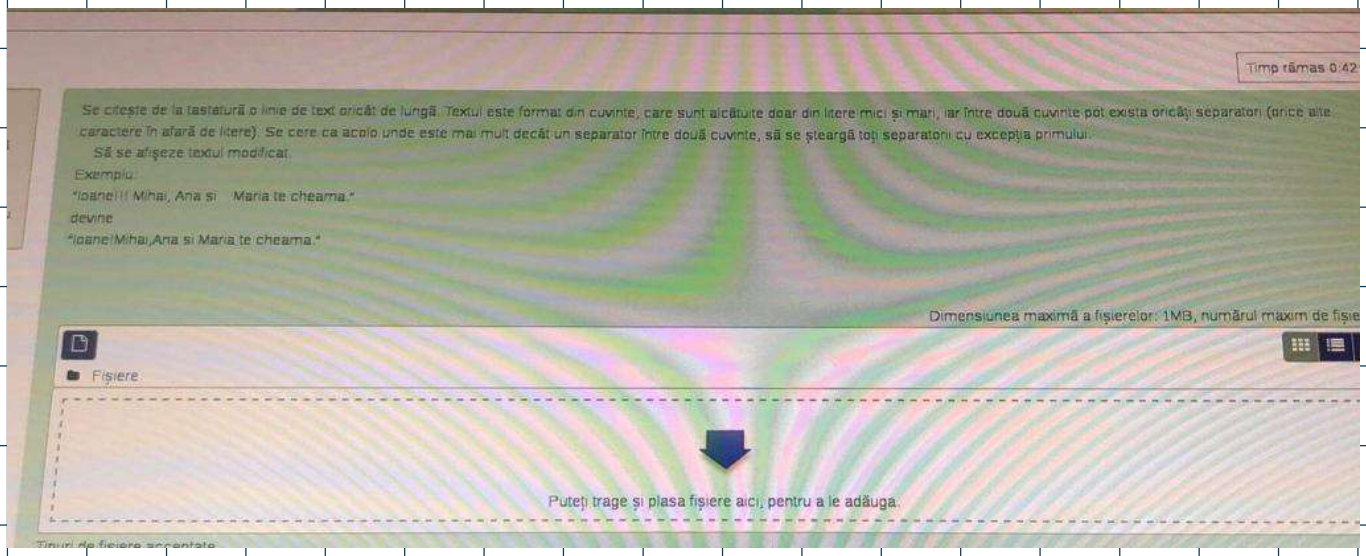
```
wget http://staff.cs.upt.ro/~valy/pt/name_list.txt
```

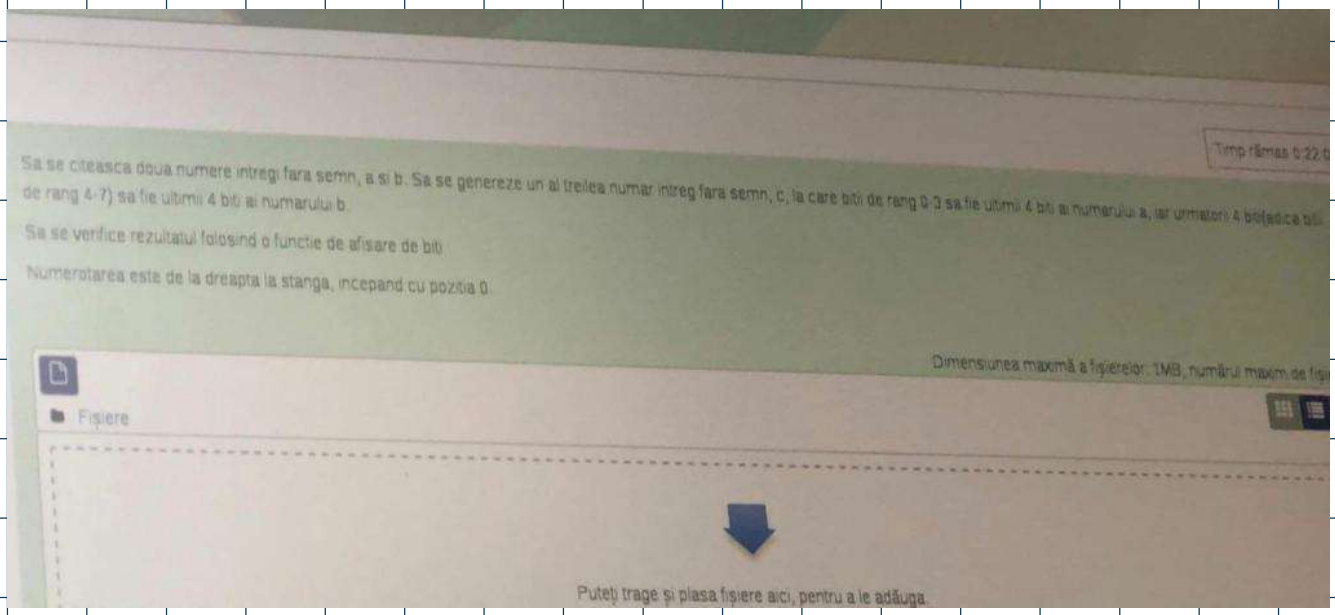
Programul va folosi functiile necesare pentru operarea cu fișiere text oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard. Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks.Se notează și elemente de eficiență. Se consideră ca fiind necunoscută dimensiunea fișierului de intrare. Pentru sortare se poate adapta funcția Bubble Sort disponibilă în secțiunea Resurse Laborator.

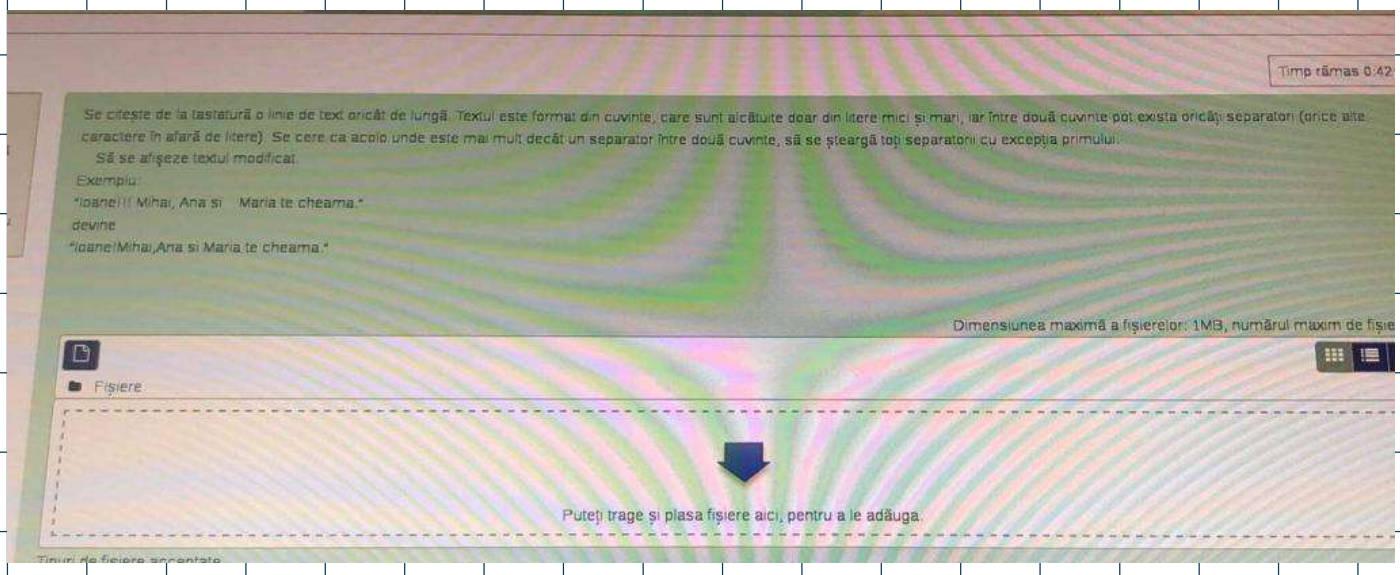
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

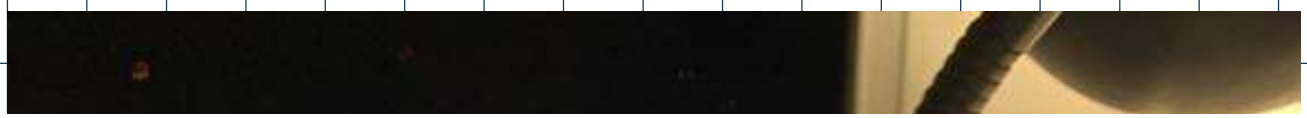
se poate adapta funcția Bubble Sort disponibilă în secțiunea Resurse Laborator.

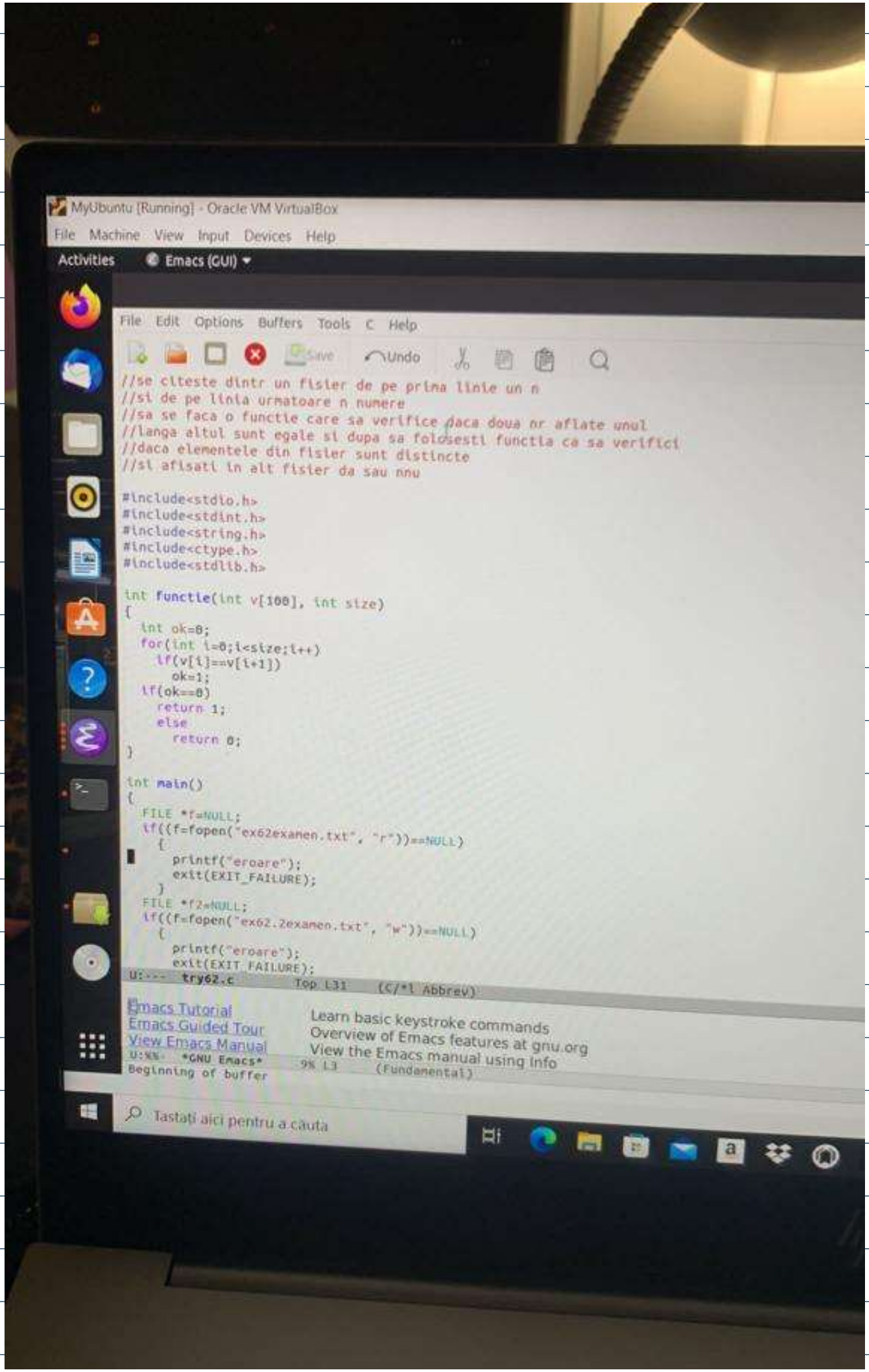
Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

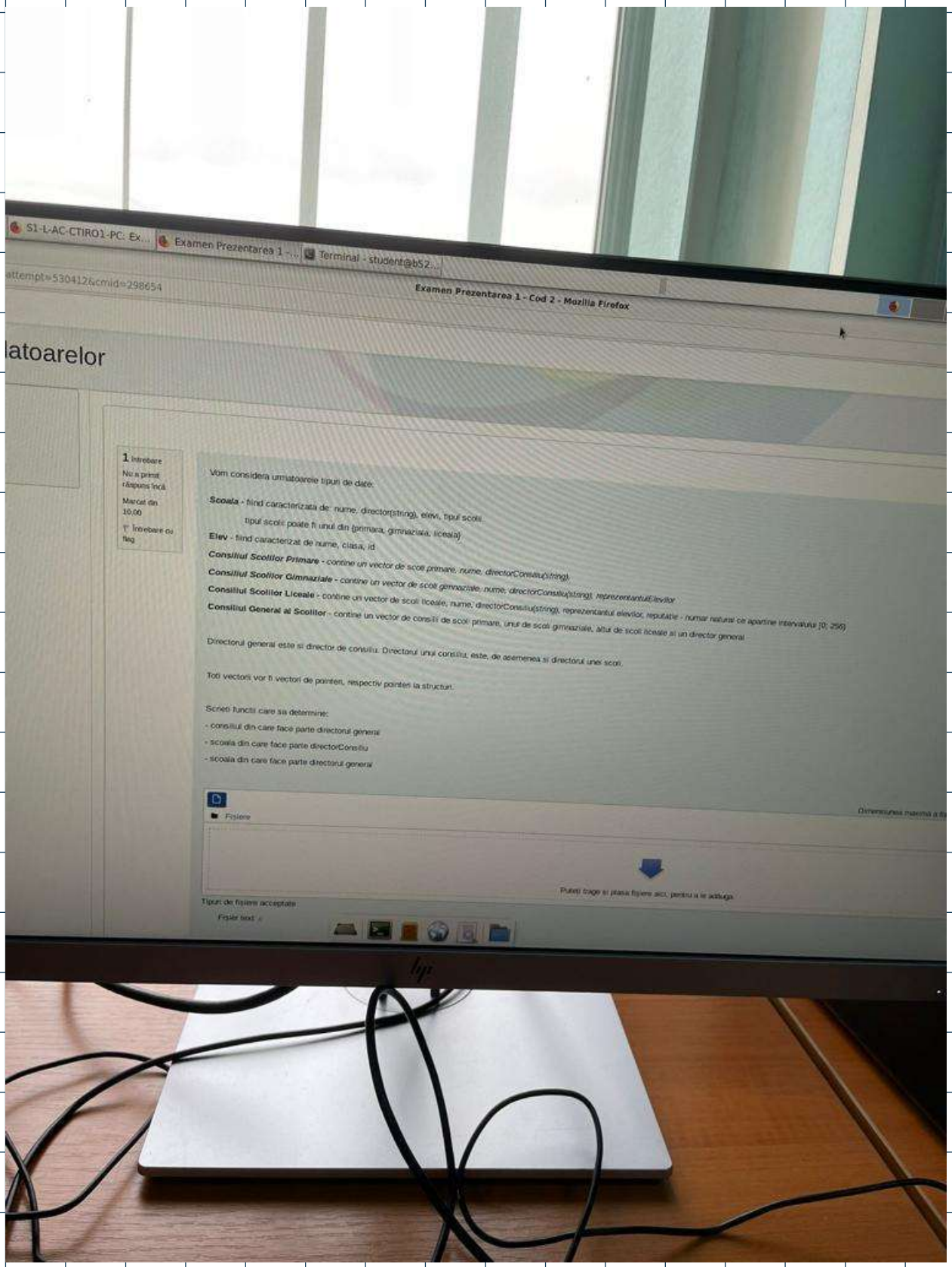


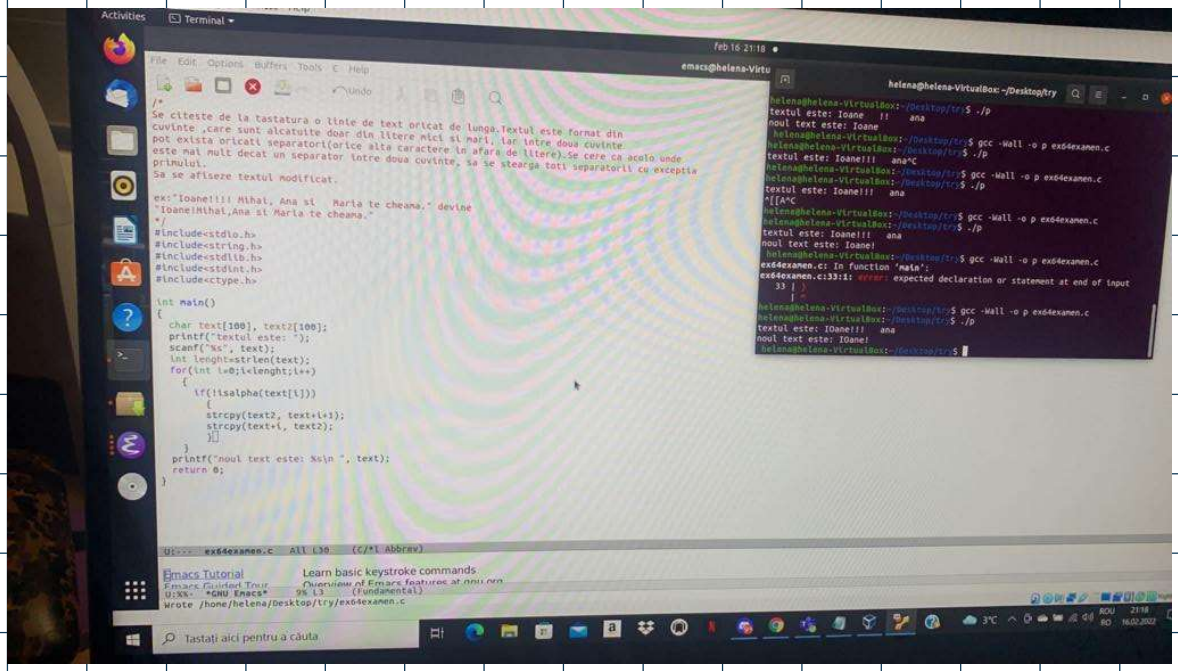







MyUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Emacs (GUI)
File Edit Options Buffers Tools C Help
Save
Undo
//se citeste dintr un fisier de pe prima linie un n
//si de pe linia urmatoare n numere
//sa se faca o functie care sa verifice daca doua nr aflate unul
//langa altul sunt egale si dupa sa folosesti functia ca sa verifici
//daca elementele din fisier sunt distincte
//si afisati in alt fisier da sau nnu
#include<stdio.h>
#include<stdint.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
int functie(int v[100], int size)
{
int ok=0;
for(int i=0;i<size;i++)
if(v[i]==v[i+1])
ok=1;
if(ok==0)
return 1;
else
return 0;
}
int main()
{
FILE *f=NULL;
if((f=fopen("ex62examen.txt", "r"))==NULL)
{
printf("eroare");
exit(EXIT_FAILURE);
}
FILE *f2=NULL;
if((f=fopen("ex62.2examen.txt", "w"))==NULL)
{
printf("eroare");
exit(EXIT_FAILURE);
U:--- try62.c Top L31 (C/*l Abbrev)
Emacs Tutorial
Emacs Guided Tour
View Emacs Manual
Learn basic keystroke commands
Overview of Emacs features at gnu.org
View the Emacs manual using Info
U:%%- *GNU Emacs* 9% L3 (Fundamental)
Beginning of buffer
Tastați aici pentru a căuta


S1-L-AC-CTIRO1-PC: Ex...
Examen Prezentarea 1 - ...
Terminal - student@b52...
attempt=530412&cmid=298654
Examen Prezentarea 1 - Cod 2 - Mozilla Firefox
latoarelor
1 întrebare
Nu a primit răspuns încă
Marcat din 10.00
Vom considera următoarele tipuri de date:
Scoala - fiind caracterizata de: nume, director(string), elevi, tipul scolii
tipul scolii poate fi unul din {primara, gimnaziala, liceala}
Elev - fiind caracterizat de nume, clasa, id
Consiliul Scolilor Primare - contine un vector de scoli primare, nume, directorConsiliu(string).
Consiliul Scolilor Gimnaziale - contine un vector de scoli gimnaziale, nume, directorConsiliu(string), reprezentantulElevilor
Consiliul Scolilor Liceale - contine un vector de scoli liceale, nume, directorConsiliu(string), reprezentantul elevilor, reputatie - numar natural ce apartine intervalului [0; 256)
Consiliul General al Scolilor - contine un vector de consilii de scoli: primare, unul de scoli gimnaziale, altul de scoli liceale si un director general
Directorul general este si director de consiliu. Directorul unui consiliu, este, de asemenea si directorul unei scoli.
Toti vectorii vor fi vectori de pointeri, respectiv pointeri la structuri.
Scrieti functii care sa determine:
- consiliul din care face parte directorul general
- scoala din care face parte directorConsiliu
- scoala din care face parte directorul general
Fișiere
Puteți trage și plasa fișiere aici, pentru a le adăuga
Tipuri de fișiere acceptate
Fișier text



Timp rămas 1:04:35

Sa se scrie un program carea va prelua ca prim argument calea către un fișier ce va fi interpretat în mod binar în care se consideră că există un număr necunoscut de numere întregi pe 2 bytes fără semn. Programul va citi acest fișier si va stoca numerele într-un tablou alocat dinamic, il va sorta apoi în ordine descrescătoare și il va scrie apoi in format text, câte un număr pe linie, într-un fișier text a cărui cale va fi primită de program ca și al doilea argument.

Programul se poate testa cu un fișier ce se poate descărca folosind următoarea comanda:

```
wget http://staff.cs.upt.ro/~valy/pt/integers.bin
```

Programul va folosi functiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard.Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks. Pentru sortare se poate folosi functia Bubble Sort din sectiunea Resurse Laborator.

Timp rămas 1:02:45

Să se scrie o funcție cu urmatorul antent

```
char *dec2text(cons char *text_dec);
```

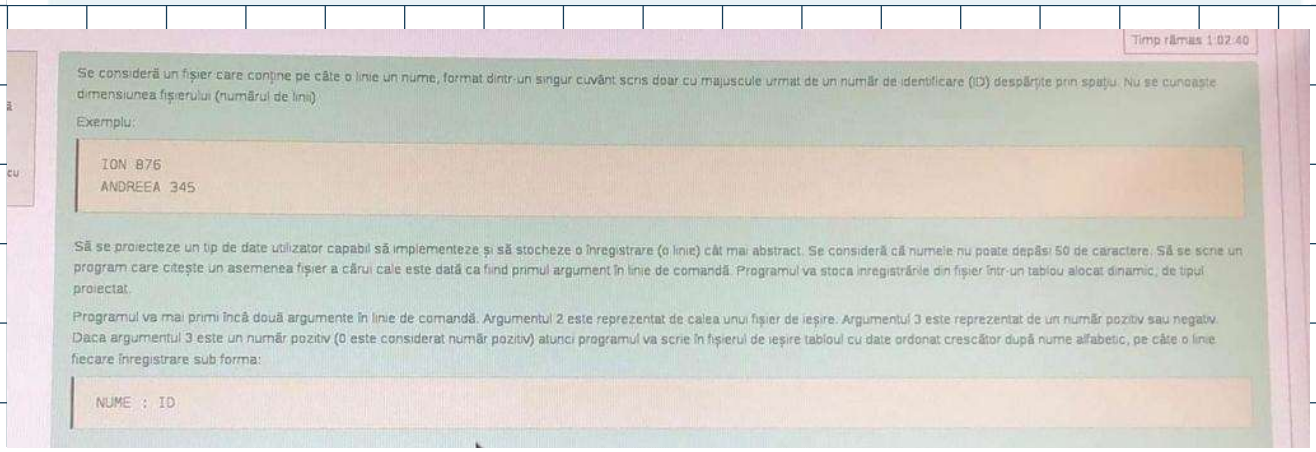
Funcția primește ca argument un string ce conține doar cifre zecimale, trei câte tre, ce reprezintă valoarea în ASCII zecimal a unui caracter și retrunează string-ul decodificat transformând fiecare grup de către 3 cifre zecimale (ce sunt în format text) în caracterul corespunzător conform reprezentării ASCII.

Exemplu: string-ul text_dec = "097098099" funcția va retuna string-ul "abc".

Se va testa funcția printr-un program ce va citi linii de text de la intrarea standard pana la EOF. Pentru fiecare linie citita programul va afișa apoi string-ul procesat folosind funcția dezvoltată anterior.

Pentru testare se recomandă utilizarea redirectării intrării standard dintr-un fișier. Un astfel de fișier se poate descărca folosind următoarea comanda:

```
wget http://staff.cs.upt.ro/~valy/pt/decalines.txt
```



fiecare înregistrare sub forma:

```
NUME : ID
```

Dacă argumentul 3 este un număr negativ atunci programul va scrie în fișierul de ieșire tabloul cu date ordonat crescător după ID sub forma:

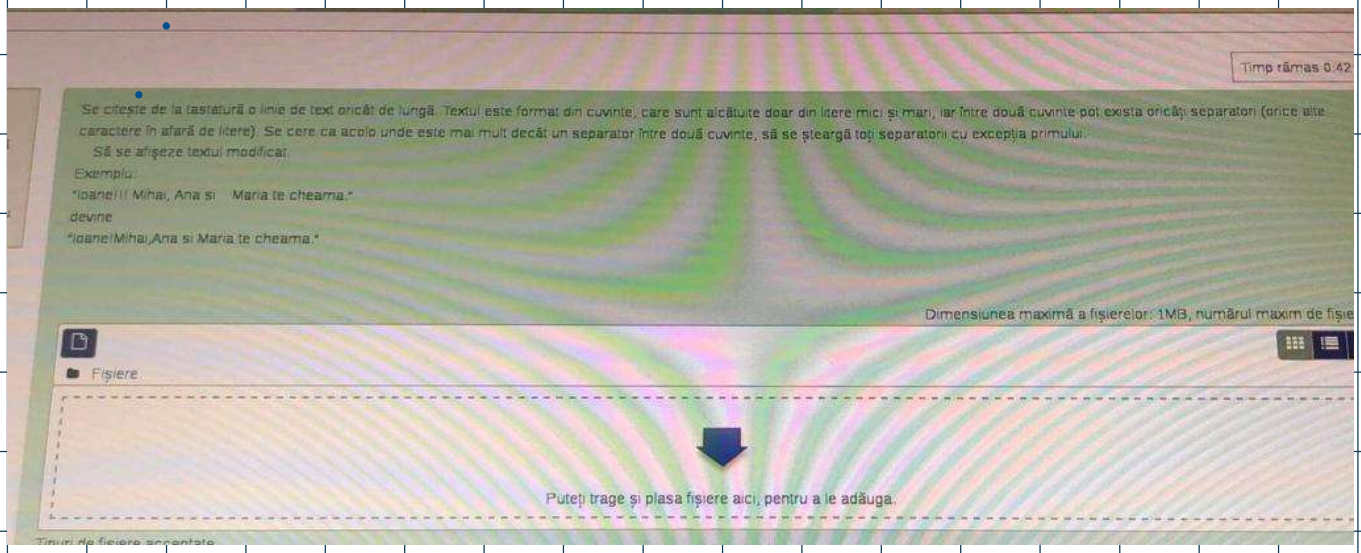
```
ID : NUME
```

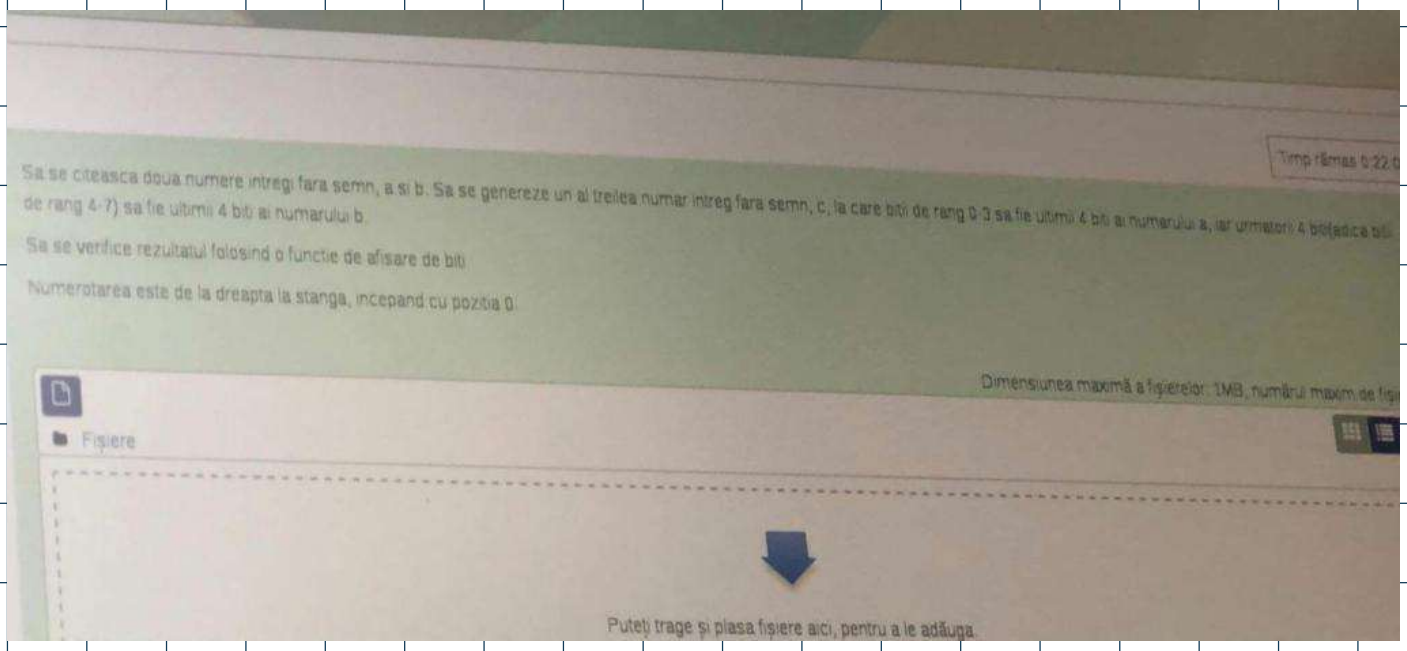
Pentru testarea programului se poate folosi un fișier ce se poate descărca folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/name_list.txt
```

Programul va folosi functiile necesare pentru operarea cu fișiere text oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard. Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks.Se notează și elemente de eficiență. Se consideră ca fiind necunoscută dimensiunea fișierului de intrare. Pentru sortare se poate adapta funcția Bubble Sort disponibilă în secțiunea Resurse Laborator.

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1





Sa se scrie un program carea va prelua ca prim argument calea către un fișier ce va fi interpretat în mod binar în care se consideră că există un număr necunoscut de numere întregi pe 2 bytes fără semn. Programul va citi acest fișier si va stoca numerele într-un tablou alocat dinamic, il va sorta apoi în ordine descrescătoare și il va scrie apoi in format text, câte un număr pe linie, într-un fișier text a cărui cale va fi primită de program ca și al doilea argument.

Programul se poate testa cu un fișier ce se poate descărca folosind următoarea comanda:

```
wget http://staff.cs.upt.ro/~valy/pt/integers.bin
```

Programul va folosi functiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard.Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks. Pentru sortare se poate folosi functia Bubble Sort din sectiunea Resurse Laborator.

Programul va folosi functiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard.Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks. Pentru sortare se poate folosi functia Bubble Sort din sectiunea Resurse Laborator.

Timp rămas 1:02:45

Să se scrie o funcție cu urmatorul antent

```
char *dec2text(cons char *text_dec);
```

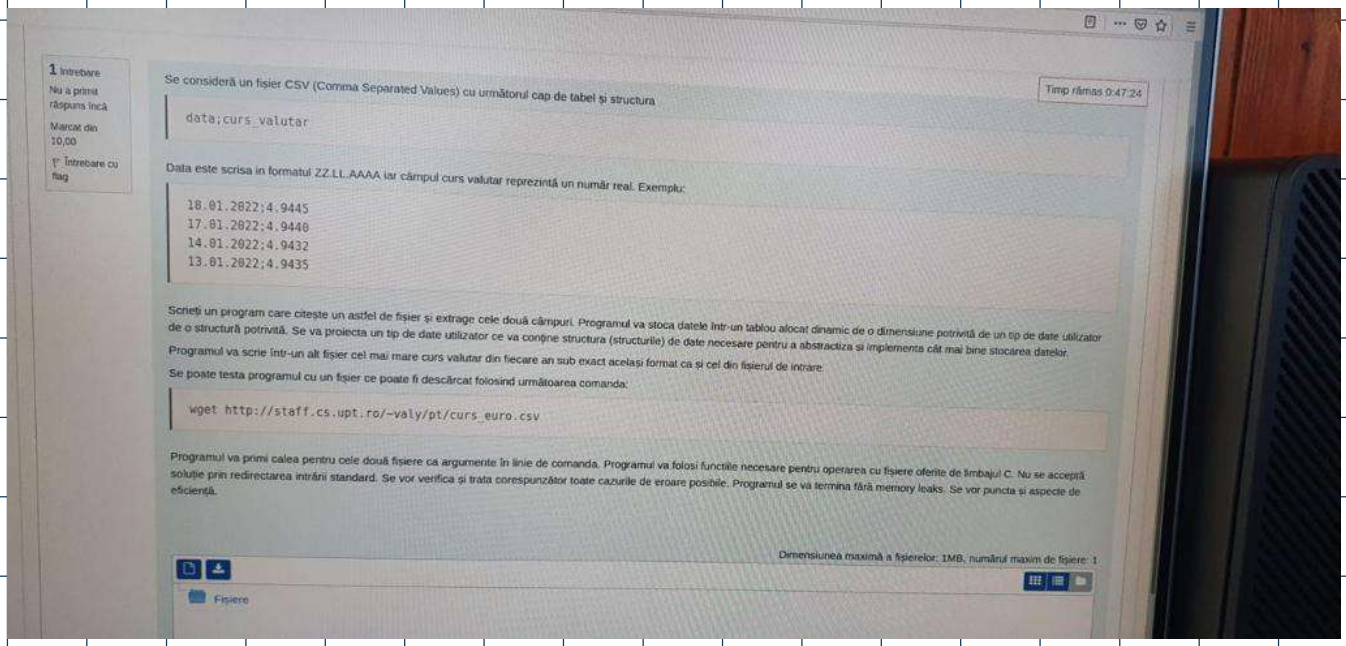
Funcția primește ca argument un string ce conține doar cifre zecimale, trei câte tre, ce reprezintă valoarea în ASCII zecimal a unui caracter și retrunează string-ul decodificat transformând fiecare grup de către 3 cifre zecimale (ce sunt în format text) în caracterul corespunzător conform reprezentării ASCII.

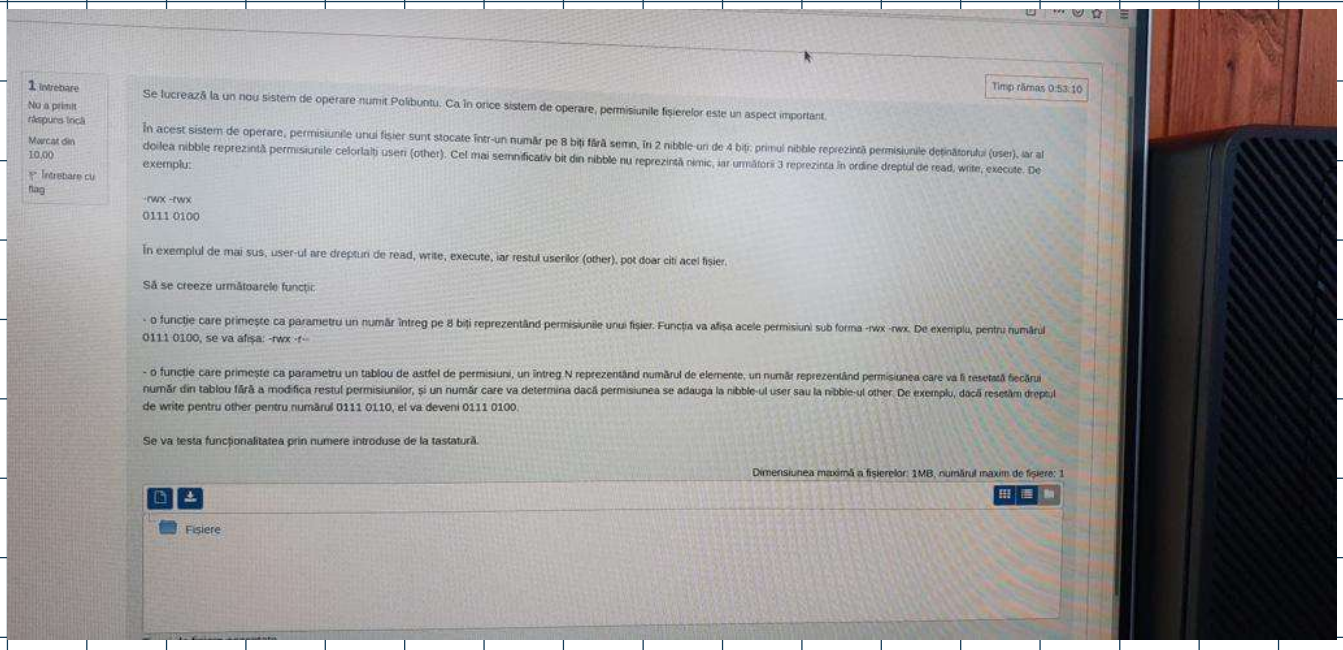
Exemplu: string-ul text_dec = "097098099" funcția va retuna string-ul "abc".

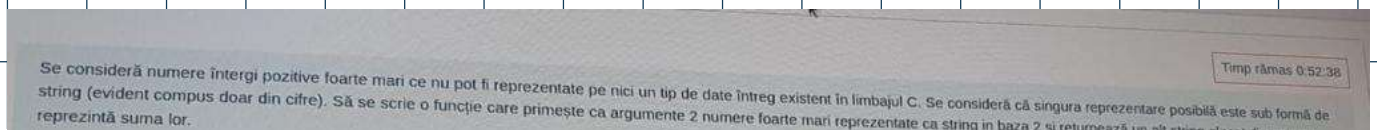
Se va testa funcția printr-un program ce va citi linii de text de la intrarea standard pana la EOF. Pentru fiecare linie citita programul va afișa apoi string-ul procesat folosind funcția dezvoltată anterior.

Pentru testare se recomandă utilizarea redirectării intrării standard dintr-un fișier. Un astfel de fișier se poate descărca folosind următoarea comanda:

```
wget http://staff.cs.upt.ro/~valy/pt/decalines.txt
```







Timp rămas 0:52:38

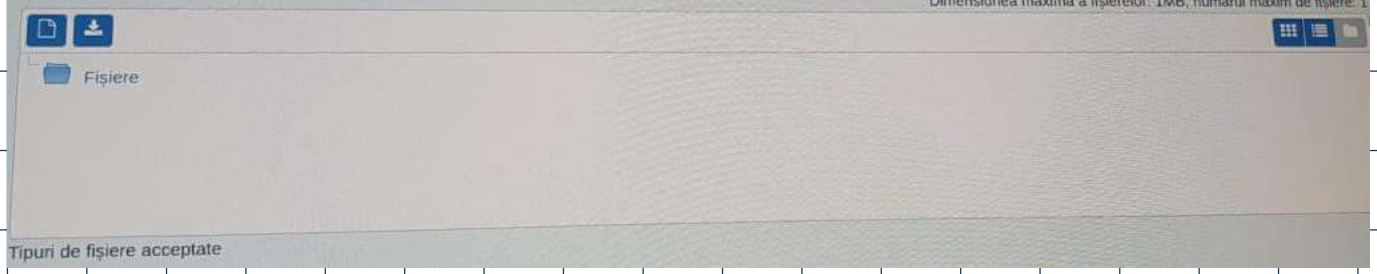
Se consideră numere întergi pozitive foarte mari ce nu pot fi reprezentate pe nici un tip de date întreg existent în limbajul C. Se consideră că singura reprezentare posibilă este sub formă de string (evident compus doar din cifre). Să se scrie o funcție care primește ca argumente 2 numere foarte mari reprezentate ca string in baza 2 și returnează un alt string alocat dinamic ce reprezintă suma lor.

```
char *add_big_numbers_binary(char *nr1, char *nr2);
```

Să se testeze această funcție într-un program cu date citite de la tastatură. Se vor citi de la tastatură 2 linii ce reprezintă fiecare câte un număr foarte mare reprezentat ca string.

Exemplu de string ce conține un număr in baza 2: "00100011101001111101010010001"

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1



Tipuri de fișiere acceptate

Timp rămas 1:02:15

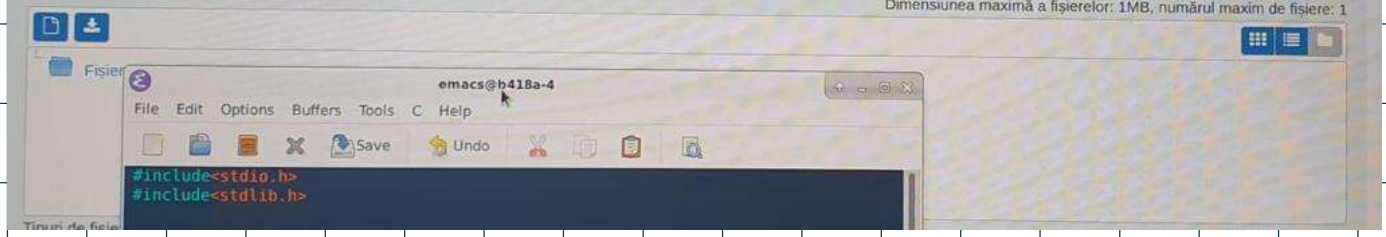
Sa se determine suma parametrilor din linia de comanda.

Daca suma este un numar prim, atunci se va scrie intr-un fisier, primele 1 milion de numere prime, cate unul pe linie.

In caz contrar, se vor citi din fisierul impar.txt o serie de numere impare despartite printr-un singur spatiu iar din fisierul par.txt, o serie de numere pare despartite printr-un singur spatiu. Se va scrie intr-un alt fisier, pe prima linie numerele impare sortate crescator, despartite printr-un singur spatiu, iar pe a doua linie, numerele pare sortate descrescator, despartite printr-un singur spatiu.

Se va scrie un algoritm eficient din punctul de vedere al timpului de executie.

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1



1 întrebare
Nu a primit răspuns încă
Marcat din 10,00
Întrebare cu flag

Timp rămas 0:47:24

Se consideră un fișier CSV (Comma Separated Values) cu următorul cap de tabel și structura

```
data;curs_valutar
```

Data este scrisa in formatul ZZ.LL.AAAA iar câmpul curs valutar reprezintă un număr real. Exemplu:

```
18.01.2022;4.9445
17.01.2022;4.9440
14.01.2022;4.9432
13.01.2022;4.9435
```

Scrieți un program care citește un astfel de fișier și extrage cele două câmpuri. Programul va stoca datele într-un tablou alocat dinamic de o dimensiune potrivită de un tip de date utilizator de o structură potrivită. Se va proiecta un tip de date utilizator ce va conține structura (structurile) de date necesare pentru a abstractiza și implementa cât mai bine stocarea datelor.

Programul va scrie într-un alt fișier cel mai mare curs valutar din fiecare an sub exact același format ca și cel din fișierul de intrare.

Se poate testa programul cu un fișier ce poate fi descărcat folosind următoarea comanda:

```
wget http://staff.cs.upt.ro/~valy/pt/curs_euro.csv
```

Programul va primi calea pentru cele două fișiere ca argumente în linie de comanda. Programul va folosi funcțiile necesare pentru operarea cu fișiere oferite de limbajul C. Nu se acceptă soluție prin redirectarea intrării standard. Se vor verifica și trata corespunzător toate cazurile de eroare posibile. Programul se va termina fără memory leaks. Se vor puncta și aspecte de eficiență.

Dimensiunea maximă a fișierelor: 1MB, numărul maxim de fișiere: 1

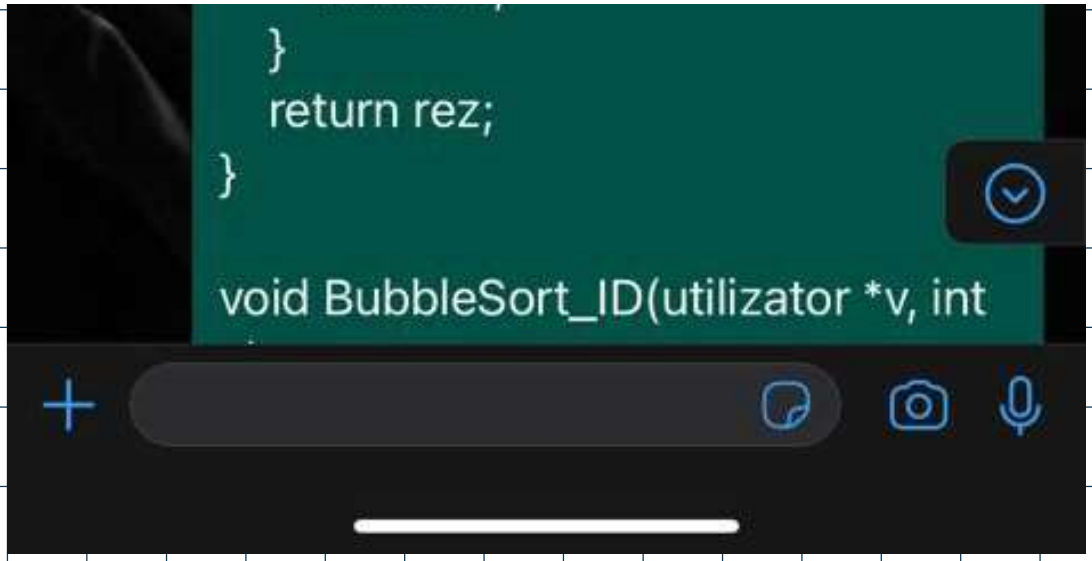
09:34 LTE

2 Bettina

```
#define CHUNK 8

typedef struct
{
   char nume[51];
   int ID;
} utilizator;

utilizator *read(FILE *in, int *size)
{
   utilizator *rez = NULL;
   utilizator aux;
   int index = 0;
   while (fscanf(in, "%s %d",
aux.nume, &aux.ID) == 2)
   {
      if (index >= *size)
      {
         *size += CHUNK;
         if ((rez = realloc(rez, size
sizeof(utilizator))) == NULL)
            exit(EXIT_FAILURE);
      }
      rez[index] = aux;
      index++;
   }
```


}
return rez;
}
void BubbleSort_ID(utilizator *v, int