



Piscina C

Ziua 12

Staff 42 piscina@academyplus.ro

Sumar: Acest document este subiectul zilei 12 a piscinei C din cadrul Academy+Plus.

Cuprins

I	Instructiuni	2
II	Preambul	4
III	Exercitiu 00 : display_file	5
IV	Exercitiu 01 : cat	6
V	Exercitiu 02 : tail	7
VI	Exercitiu 03 : hexdump	8
VII	Exercitiu 04 : last	9

Capitolul I

Instructiuni

- Utilizati doar aceste pagini ca referinta; nu plecati urechea la zgomotul de pe coridor.
- Subiectul se poate schimba cu cel mult o ora inainte de incepere.
- Fiti atenti la drepturile pe care le aveti asupra fisierelor si directoarelor.
- Trebuie sa urmati procedurile de parcurgere pentru toate exercitiile voastre.
- Exerciitiile voastre vor fi corectate de colegii vostri de piscina.
- Pe langa colegii vostri, veti fi corectati de un program numit Moulinette.
- Aplicatia Moulinette este foarte stricta in notare. Ea este total automatizata. Este imposibil sa comentati in legatura cu nota primita. Fiti foarte rigurosi pentru a evita surprizele.
- Moulinette nu e foarte desteapta. Ea nu poate intelege codul care nu respecta Standardele de scriere a codului (Norme).
- Utilizarea unei functii interzise este un caz de inselaciune (trisare). Toate aceste cazuri sunt sanctionate cu nota -42.
- Daca `ft_putchar()` este o functie valida, veti compila fisierul `ft_putchar.c`.
- Nu trebuie sa creati o functie `main()` decat atunci cand vi se cere sa scrieti un program.
- Exerciitiile sunt strict ordonate de la cele simple spre cele complexe. In nici un caz nu vom lua in considerare un exercitiu complex rezolvat daca unul anterior, mai simplu, nu a fost rezolvat perfect.
- Aplicatia Moulinette se compileaza cu flag-urile: `-Wall -Wextra -Werror`.
- Daca programul vostru nu se compileaza, veti primi nota 0.

- Nu lasati in directorul de lucru niciun fisier, altul decat cele specificate de enuntul exercitiului.
- Aveti intrebari? Intrebat-l pe vecinul din dreapta. Daca nu, incercati la cel din stanga.
- Manualele voastre de referinta sunt Google / man / Internet /
- Puteti folosi forumul de pe Intranet pentru discutii legate de Piscina!
- Cititi cu atentie exemplele. Va pot oferi informatii suplimentare pentru elementele neclare din enunt...
- Reflectati la asta. Aveti mare grija!

Capitolul II

Preambul

Iată rețeta pentru Carne de urs în bere:

Timp de preparare: în jur de 30 de minute

Timp de gătire: 90 de minute

Ingrediente (Pentru 4 persoane):

- 1kg de carne de urs (spată, în mod ideal)
- 5 cepe mari
- câteva ramuri de cimbru
- 5 foi de dafin
- 2 sfeclă crude, curățate și tăiate în sferturi
- 800 ml de bere brună
- un pumn mare de făină
- ulei
- sare
- piper


Preparare:

- Tăiați carnea cubulețe
- Prăjiți carnea într-o tigaie cu puțin ulei
- Adăugați sare și piper
- Dați carnea la o parte
- În aceeași tigaie, prăjiți ceapa și cimbrul, dafinul și sfecla la foc mediu
- Presărați făină, apoi diluați cu bere
- Adăugați carnea
- Acoperiți, apoi coaceți în cuptor la 180 de grade pentru aproximativ o oră.
- Se servește cald!

Subiectul de azi nu are nimic de a face cu ursul în bere.

Capitolul III

Exercitiu 00 : display_file


	Exercitiu: 00
display_file	
Director de lucru: <i>ex00/</i>	
Fisier(e) de iesire: Makefile, si fisierele programului vostru	
Functii autorizate: close, open, read, write	
Observatii: n/a	

- Scrieti un program numit `ft_display_file` care afiseaza la iesirea standard doar continutul fisierului transmis ca argument.
- Directorul de lucru va avea `Makefile` cu o regula `all`, o regula `clean`, si o regula `fclean`. Fisierul binar se va numi `ft_display_file`.
- Functia `malloc` e interzisa. Puteti face exercitiul doar declarand un tablou de dimensiune fixa.
- Toate fisierele transmise ca parametru vor fi valide.
- Mesajele de eroare trebuie sa fie afisate pe iesirea care le-a fost rezervata (iesirea de eroare standard).

```
$> ./ft_display_file
File name missing.
$> ./ft_display_file Makefile
*contenu du Makefile*
$> ./ft_display_file Makefile display_file.c
Too many arguments.
$>
```

Capitolul IV

Exercitiu 01 : cat


	Exercitiu: 01
cat	
Director de lucru: <i>ex01/</i>	
Fisier(e) de iesire: Makefile, si fisierele programului vostru	
Functii autorizate: close, open, read, write	
Observatii: n/a	

- Scrieti un program numit `ft_cat` care realizeaza aceeasi actiune ca la comanda `cat` de sistem.
- Nu trebuie sa va ocupati de optiuni.
- Directorul de lucru va contine un Makefile cu o regula `all`, o regula `clean`, si o regula `fclean`.
- Puteti utiliza variabila `errno` (vedeti man pentru `Errno`).
- Puteti face exercitiul doar declarand un tablou de dimensiune fixa. Acest tablou va avea o dimensiune limitata la aproximativ 30 ko. Pentru a putea testa aceasta limitare utilizati comanda `limit` din shell.

```
$> limit stacksize 32
$> limit stacksize
stacksize 32 kbytes
$>
```

Capitolul V


Exercitiu 02 : tail

	Exercitiu: 02
tail	
Director de lucru: <i>ex02/</i>	
Fisier(e) de iesire: <i>Makefile</i> , si fisierele programului vostru	
Functii autorizate: <i>close, open, read, write, malloc, free</i>	
Observatii: <i>n/a</i>	

- Scrieti un program numit `ft_tail` care realizeaza aceeasi actiune ca la comanda `tail` de sistem, dar care primeste cel putin un fisier ca argument.
- Va trebui sa utilizati doar optiunea `-c`; aceasta optiune va fi utilizata in toate testele.
- Directorul de lucru va contine un `Makefile` cu o regula `all`, o regula `clean`, si o regula `fclean`.
- Veti putea utiliza variabila `errno` (vedeti `man` pentru `errno`).

Capitolul VI


Exercitiu 03 : hexdump

	Exercitiu: 03
hexdump	
Director de lucru: <code>ex03/</code>	
Fisier(e) de iesire: <code>Makefile</code> , si fisierele programului vostru	
Functii autorizate: <code>close</code> , <code>open</code> , <code>read</code> , <code>write</code> , <code>malloc</code> , <code>free</code>	
Observatii: <code>n/a</code>	

- Scrieti un program numit `ft_hexdump` care va realiza aceeasi actiune ca la comanda `hexdump` de sistem.
- Nu puteti sa utilizati decat optiunea `-C`.
- Directorul de lucru va avea un `Makefile` cu o regula `all`, o regula `clean`, si o regula `fclean`.
- Veti putea utiliza variabila `errno` (vedeti `man` pentru `errno`).

Capitolul VII

Exercitiu 04 : last

	Exercitiu: 04
last	
Director de lucru: <i>ex04/</i>	
Fisier(e) de iesire: <i>Makefile</i> , si fisierele programului vostru	
Functii autorizate: <i>close, open, read, write, malloc, free</i>	
Observatii: <i>n/a</i>	

- Scrieti un program numit `ft_last` care va realiza aceeasi actiune ca la comanda `last` de sistem, fara nicio optiune.
- Directorul de lucru va contine un fisier `Makefile` cu o regula `all`, o regula `clean`, si o regula `fclean`.
- Aveti dreptul de a `include` doar `<utmp.h>`, `<time.h>`, `<errno.h>` si `<sys/types.h>`
- Nu puteti utiliza functiile `ctime()`, `asctime()`, `stat()`, etc.