



# Piscina C

Ziua 03

Staff Academy+Plus [contact@academyplus.ro](mailto:contact@academyplus.ro)

*Sumar: Acest document este subiectul zilei a 03 a piscinei C din cadrul Academy+Plus.*

# Cuprins

I	Instructiuni	2
II	Preambul	4
III	Exercitiu 00 : ft_ft	5
IV	Exercitiu 01 : ft_ultimate_ft	6
V	Exercitiu 02 : ft_swap	7
VI	Exercitiu 03 : ft_div_mod	8
VII	Exercitiu 04 : ft_ultimate_div_mod	9
VIII	Exercitiu 05 : ft_putstr	10
IX	Exercitiu 06 : ft_strlen	11
X	Exercitiu 07 : ft_strrev	12
XI	Exercitiu 08 : ft_atoi	13
XII	Exercitiu 09 : ft_sort_integer_table	14

# 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 de piscina.
- Pe langa colegi, va veti folosi 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).
- 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.
- 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.
- 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

Jocul Siropului de Perceval, din seria *Kaamelott* :

„În regulă, vă voi învăța regulile simplificate, întrucât regulile adevărate sunt stufoase. Ei bine, singurul lucru este că în mod normal acesta se joacă în trei. Nu-i nimic, ne vom descurca.

Principiul este să facem valori. Astfel, să zicem că suntem trei, există prin urmare trei valori care trebuie distribuite. Să zicem, sirop de opt, sirop de paisprezece și sirop de douăzeci și unu. Nu trebuie să vă preocupe siropurile imediat. Ce trebuie să înțelegeți în primă instanță sunt valorile. Dacă lansați o valoare la început de tur, vom pune sirop de opt, pentru a începe cu puțin, iar ceilalți au alegerea de a păstra miza sau de a relansa un sirop de paisprezece. Rotația se face în sensul valorilor. De aceea este important să înțelegeți bine sistemul de valori; după aceea, totul merge de la sine.

În regulă, să spunem că eu încep cu un sirop de opt.

Dacă în runda anterioară ați sorbit dvs., se va merge în direcția dvs. Prin urmare, spuneți „pas”, spuneți „pauză-sirop” sau poate doriți să relansați și anunțați un sirop de paisprezece. Când se încep anunțurile, al doilea jucător nu are voie să spună „pas”. Puteți relansa fie un sirop de douăzeci și unu, fie abandonați runda și spuneți „strat-sirop” fie „sirop-Jeannot”, în funcție de regiuni. Apoi, de acolo sirop de mere, la douăzeci și unu facem runda în patru ture până când unul dintre noi ia o sorbire.


Pentru a câștiga nu există decât trei posibilități: fie vă faceți propriul sirop de opt, spuneți „sirop frumos” și renumărăm, fie vă faceți propriul sirop de paisprezece, spuneți „sirop frumos, sirop câștigător” și adăugați jumătate, fie faceți propriul sirop de douăzeci și unu și spuneți „sirop frumos, jumătate de sirop, înghițitură, sirop câștigător, sirop clopot, cagulă alpină, sirop cu gust bun”.

În mod normal, se joacă cu cărți dar dacă nu aveți decât zaruri, puteți juca și cu zaruri pentru că ceea ce contează sunt valorile.

Cel puțin unul dintre exercitiile de mai jos nu are legatura cu jocul de Sirop.

# Capitolul III

## Exercitiu 00 : ft\_ft


	Exercitiu: 00
ft_ft	
Director de lucru: <i>ex00/</i>	
Fisier(e) de iesire: <b>ft_ft.c</b>	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care ia ca parametru un pointer spre un int si care atribuie valoarea 42 intregului.
- Ea trebuie sa aiba prototipul urmator:

```
void      ft_ft(int *nbr);
```

# Capitolul IV

## Exercitiu 01 : ft\_ultimate\_ft


	Exercitiu: 01
	ft_ultimate_ft
	Director de lucru: ex01/
	Fisier(e) de iesire: ft_ultimate_ft.c
	Functii autorizate: Niciuna
	Observatii: n/a

- Scrieti o functie care ia ca parametru un pointer la pointer la pointer la pointer la pointer la pointer la pointer la pointer la int si care atribuie valoarea 42 intregului.
- Ea trebuie sa aiba prototipul urmator:

```
void ft_ultimate_ft(int *****nbr);
```

# Capitolul V

## Exercitiu 02 : ft\_swap

	Exercitiu: 02
ft_swap	
Director de lucru: ex02/	
Fisier(e) de iesire: ft_swap.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care schimba doua numere intregi ale caror adrese sunt date ca parametri.
- Ea trebuie sa aiba prototipul urmator:

```
void    ft_swap(int *a, int *b);
```



# Capitolul VI

## Exercitiu 03 : ft\_div\_mod

	Exercitiu: 03
	ft_div_mod
Director de lucru: ex03/	
Fisier(e) de iesire: ft_div_mod.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie `ft_div_mod` care are prototipul urmator:

```
void ft_div_mod(int a, int b, int *div, int *mod);
```

- Aceasta functie imparte cei doi parametri `a` si `b` si stocheaza rezultatul intr-un intreg spre care pointeaza `div`.  
Ea stocheaza de asemenea restul impartirii `a` si `b` intr-un intreg spre care pointeaza `mod`.

# Capitolul VII

## Exercitiu 04 : ft\_ultimate\_div\_mod

	Exercitiu: 04
	ft_ultimate_div_mod
Director de lucru: ex04/	
Fisier(e) de iesire: ft_ultimate_div_mod.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie `ft_ultimate_div_mod` care are prototipul urmator:

```
void ft_ultimate_div_mod(int *a, int *b);
```

- Aceasta functie imparte doi intregi spre care pointeaza **a** si **b**.  
Rezultatul impartirii este stocat intr-un intreg spre care pointeaza **a**.  
Restul impartirii este stocat intr-un intreg spre care pointeaza **b**.

# Capitolul VIII

## Exercitiu 05 : ft\_putstr


	Exercitiu: 05
ft_putstr	
Director de lucru: ex05/	
Fisier(e) de iesire: ft_putstr.c	
Functii autorizate: ft_putchar	
Observatii: n/a	

- Scrieti o functie care afiseaza pe ecran caracterele unui sir, unul cate unu.
- Adresa primului caracter al sirului este continuta in pointerul care este dat ca parametru functiei.
- Ea trebuie sa aiba prototipul urmator:

```
void    ft_putstr(char *str);
```

# Capitolul IX

## Exercitiu 06 : ft\_strlen


	Exercitiu: 06
ft_strlen	
Director de lucru: <i>ex06/</i>	
Fisier(e) de iesire: <b>ft_strlen.c</b>	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care contorizeaza numărul de caractere dintr-un sir de caractere si returnează numărul găsit.
- Ea trebuie sa aiba prototipul urmator:

```
int      ft_strlen(char *str);
```

# Capitolul X

## Exercitiu 07 : ft\_strrev

	Exercitiu: 07
ft_strrev	
Director de lucru: ex07/	
Fisier(e) de iesire: ft_strrev.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care inverseaza un sir de caractere.
- Ea va trebui sa returneze sirul inversat.
- Ea trebuie sa aiba prototpul urmator:


```
char *ft_strrev(char *str);
```

- Exemplu:

```
a => a  
ab => ba  
abcde => edcba
```

# Capitolul XI

## Exercitiu 08 : ft\_atoi


	Exercitiu: 08
ft_atoi	
Director de lucru: <i>ex08/</i>	
Fisier(e) de iesire: <b>ft_atoi.c</b>	
Functii autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei `atoi` (man `atoi`).
- Ea va trebui sa aiba prototipul urmator:

```
int ft_atoi(char *str);
```

# Capitolul XII

## Exercitiu 09 : ft\_sort\_integer\_table

	Exercitiu: 09
	ft_sort_integer_table
Director de lucru: ex09/	
Fisier(e) de iesire: ft_sort_integer_table.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care sorteaza un tabel de numere intregi in ordine crescatoare.
- Parametri acestei functii sunt un pointer spre (primul) intreg si numarul de intregi din tablou.
- Functia trebuie sa aiba prototipul urmator:

```
void ft_sort_integer_table(int *tab, int size);
```