



Piscina C

Ziua 05

Staff Academy+Plus contact@academyplus.ro

Sumar: Acest document este subiectul zilei a 05 a piscinei C din cadrul Academy+Plus.

Cuprins

I	Instructiuni	2
II	Preambul	4
III	Exercitiu 00 : ft_putstr	5
IV	Exercitiu 01 : ft_putnbr	6
V	Exercitiu 02 : ft_atoi	7
VI	Exercitiu 03 : ft_strcpy	8
VII	Exercitiu 04 : ft_strncpy	9
VIII	Exercitiu 05 : ft_strstr	10
IX	Exercitiu 06 : ft_strcmp	11
X	Exercitiu 07 : ft_strncmp	12
XI	Exercitiu 08 : ft_strupcase	13
XII	Exercitiu 09 : ft_strlowercase	14
XIII	Exercitiu 10 : ft_strcapitalize	15
XIV	Exercitiu 11 : ft_str_is_alpha	16
XV	Exercitiu 12 : ft_str_is_numeric	17
XVI	Exercitiu 13 : ft_str_is_lowercase	18
XVII	Exercitiu 14 : ft_str_is_uppercase	19
XVIII	Exercitiu 15 : ft_str_is_printable	20
XIX	Exercitiu 16 : ft_strcat	21
XX	Exercitiu 17 : ft_strncat	22
XXI	Exercitiu 18 : ft_strlcat	23
XXII	Exercitiu 19 : ft_strlcpy	24
XXIII	Exercitiu 20 : ft_putnbr_base	25

XXIV	Exercitiu 21 : ft_atoi_base	27
XXV	Exercitiu 22 : ft_putstr_non_printable	29
XXVI	Exercitiu 23 : ft_print_memory	30

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, 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 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? Intrebati-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!



Pentru aceasta zi < norminette > va fi lansata cu flagul `-R CheckForbiddenSourceHeader`. Si moulinette il va folosi.

Capitolul II

Preambul

Posibile utilizări ale cocainei, conform D  sencyclop  die:

Recomandată pentru sugari, cocaina poate fi folosită și la numeroase feluri de mâncare, relativ diferite unele de celelalte.


Cocaina are de asemenea, puternice calități medicale. Utilizarea sa este recomandată de OMS (Organizația Mondială a Sănătății) pentru:

- Vindecarea gripei (consumați cocaină infuzată)
- Vindecarea reumatismului
- Vindecarea bolilor rumegătoarelor (boala vacii nebune și altele)
- Vindecarea tăieturilor (amestecați cocaină cu   țet și aplicați cataplasme)
- Vindecarea sindromului Vladimir-von-Umbolt. De altfel, această boală nu a fost vreodată declarată, fără îndoială mulțumită cocainei
- O moarte lentă (amestecați cocaină cu cianură și antimoniu, apoi beți până la fund)

Cum, în realitate, cocaina este de fapt o substanță foarte periculoasă, va trebui să vă abțineți de la a consuma C.

Capitolul III

Exercitiu 00 : ft_putstr

	Exercitiu: 00
ft_putstr	
Director de lucru: ex00/	
Fisier(e) de iesire: ft_putstr.c	
Functii autorizate: ft_putchar	
Observatii: n/a	


Academy+ - Classics : Aceste exercitii sunt inevitabile chiar daca nu aduc nici un punct, este obligatoriu sa le validati pentru a accesa adevaratele exercitii ale zilei.

- Scrieti o functie care afiseaza pe ecran caracterele unui sir, unul cate unul.
- 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 IV

Exercitiu 01 : ft_putnbr

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

Academy+ - Classics : Aceste exercitii sunt inevitabile chiar daca nu aduc nici un punct, este obligatoriu sa le validati pentru a accesa adevaratele exercitii ale zilei.


- Scrieti o functie care afiseaza un numar transmis ca parametru. Functia trebuie sa fie capabila sa afiseze orice valoare de tip `int`.
- Ea trebuie sa aiba prototipul urmator:

```
void ft_putnbr(int nb);
```

- De exemplu:
 - `ft_putnbr(42)` afiseaza "42".

Capitolul V

Exercitiu 02 : ft_atoi

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


Academy+ - Classics : Aceste exercitii sunt inevitabile chiar daca nu aduc nici un punct, este obligatoriu sa le validati pentru a accesa adevaratele exercitii ale zilei.

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

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

Capitolul VI

Exercitiu 03 : ft_strcpy


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

- Reproduceti identic comportamentul functiei `strcpy` (man strcpy).
- Ea trebuie sa aiba prototipul urmator:

```
char *ft_strcpy(char *dest, char *src);
```

Capitolul VII

Exercitiu 04 : ft_strncpy


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

- Reproduceti identic comportamentul functiei `strncpy` (man `strncpy`).
- Ea trebuie sa aiba prototipul urmator:

```
char      *ft_strncpy(char *dest, char *src, unsigned int n);
```

Capitolul VIII

Exercitiu 05 : ft_strstr


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

- Reproduceti identic comportamentul functiei **strstr** (man strstr).
- Ea trebuie sa aiba prototipul urmator:

```
char *ft_strstr(char *str, char *to_find);
```

Capitolul IX

Exercitiu 06 : ft_strcmp


	Exercitiu: 06
ft_strcmp	
Director de lucru: ex06/	
Fisier(e) de iesire: ft_strcmp.c	
Funcții autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei `strcmp` (man strcmp).
- Ea trebuie sa aiba prototipul urmator:

```
int ft_strcmp(char *s1, char *s2);
```

Capitolul X

Exercitiu 07 : ft_strncmp


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

- Reproduceti identic comportamentul functiei `strncmp` (man `strncmp`).
- Ea trebuie sa aiba prototipul urmator:

```
int      ft_strncmp(char *s1, char *s2, unsigned int n);
```

Capitolul XI

Exercitiu 08 : ft_strupcase

	Exercitiu: 08
ft_strupcase	
Director de lucru: ex08/	
Fisier(e) de iesire: ft_strupcase.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care schimba caracterele fiecarui cuvant in majuscule.
- Ea trebuie sa aiba prototipul urmator:

```
char *ft_strupcase(char *str);
```

- Ea trebuie sa returneze `str`.

Capitolul XII

Exercitiu 09 : ft_strlowcase

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


- Scrieti o functie care shimba caracterele fiecarui cuvant in minuscule.
- Ea trebuie sa aiba urmatorul prototip:

```
char *ft_strlowcase(char *str);
```

- Ea trebuie sa returneze **str**.

Capitolul XIII

Exercitiu 10 : ft_strcapitalize

	Exercitiu: 10
ft_strcapitalize	
Director de lucru: ex10/	
Fisier(e) de iesire: ft_strcapitalize.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care schimba in majuscula primul caracter al fiecarui cuvant si toate celelalte litere in minuscule.
- Un cuvant este o succesiune de caractere alfanumerice.
- Ea trebuie sa aiba urmatorul prototip:

```
char *ft_strcapitalize(char *str);
```

- Ea trebuie sa returneze `str`.
- De exemplu:


```
salut, ce faci? 42cuvinte patruzeci-doi; cincizeci+si+unu
```

- Trebuie sa dea:

```
Salut, Ce faci? 42cuvinte Patruzeci-Doi; Cincizeci+Si+Unu
```

Capitolul XIV

Exercitiu 11 : ft_str_is_alpha

	Exercitiu: 11
ft_str_is_alpha	
Director de lucru: ex11/	
Fisier(e) de iesire: ft_str_is_alpha.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care returneaza 1 daca sirul de caractere transmis ca parametru nu contine decat caractere alfabetice si returneaza 0 daca acesta contine si alte tipuri de caractere.
- Ea trebuie sa aiba urmatorul prototip:

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

- Ea trebuie sa returneze 1 daca **str** este un sir vid.

Capitolul XV

Exercitiu 12 : ft_str_is_numeric

	Exercitiu: 12
ft_str_is_numeric	
Director de lucru: ex12/	
Fisier(e) de iesire: ft_str_is_numeric.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care returneaza 1 daca sirul de caractere transmis ca paramentru nu contine decat cifre si returneaza 0 daca sirul contine si alte caractere.
- Ea trebuie sa aiba urmatorul prototip:

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

- Ea trebuie sa returneze 1 daca **str** este un sir vid.

Capitolul XVI

ExerciTiu 13 : ft_str_is_lowercase

	Exercitiu: 13
ft_str_is_lowercase	
Director de lucru: ex13/	
Fisier(e) de iesire: ft_str_is_lowercase.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care returneaza 1 daca sirul de caractere transmis ca parametru nu contine decat caractere alfabetice minuscule si returneaza 0 daca sirul contine si alte caractere.
- Ea trebuie sa aiba urmatorul prototip:

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

- Ea trebuie sa returneze 1 daca `str` este un sir vid.

Capitolul XVII

Exercitiu 14 : ft_str_is_uppercase

	Exercitiu: 14
ft_str_is_uppercase	
Director de lucru: ex14/	
Fisier(e) de iesire: ft_str_is_uppercase.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care returneaza 1 daca sirul de caractere transmis ca parametru nu contine decat caractere alfabetice majuscule si returneaza 0 daca sirul contine alte caractere.
- Ea trebuie sa aiba urmatorul prototip:

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

- Ea trebuie sa returneze 1 daca **str** este un sir vid.

Capitolul XVIII

Exercitiu 15 : ft_str_is_printable

	Exercitiu: 15
ft_str_is_printable	
Director de lucru: ex15/	
Fisier(e) de iesire: ft_str_is_printable.c	
Functii autorizate: Niciuna	
Observatii: n/a	


- Scrieti o functie care returneaza 1 daca sirul de caractere transmis ca parametru nu contine decat caractere tiparibile si returneaza 0 daca sirul contine si alte caractere.
- Ea trebuie sa aiba urmatorul prototip:

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

- Ea trebuie sa returneze 1 daca **str** este un sir vid.

Capitolul XIX

Exercitiu 16 : ft_strcat


	Exercitiu: 16
ft_strcat	
Director de lucru: <i>ex16/</i>	
Fisier(e) de iesire: ft_strcat.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei **strcat** (man strcat).
- Ea trebuie sa aiba urmatorul prototip:

```
char *ft_strcat(char *dest, char *src);
```

Capitolul XX

Exercitiu 17 : ft_strncat


	Exercitiu: 17
ft_strncat	
Director de lucru: <i>ex17/</i>	
Fisier(e) de iesire: ft_strncat.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei **strncat** (man strncat).
- Ea trebuie sa aiba urmatorul prototip:

```
char *ft_strncat(char *dest, char *src, int nb);
```


Capitolul XXI

Exercitiu 18 : ft_strlcat


	Exercitiu: 18
ft_strlcat	
Director de lucru: <i>ex18/</i>	
Fisier(e) de iesire: ft_strlcat.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei **strlcat** (man strlcat).
- Ea trebuie sa aiba urmatorul prototip:

```
unsigned int ft_strlcat(char *dest, char *src, unsigned int size);
```

Capitolul XXII

Exercitiu 19 : ft_strlcpy


	Exercitiu: 19
ft_strlcpy	
Director de lucru: <i>ex19/</i>	
Fisier(e) de iesire: ft_strlcpy.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Reproduceti identic comportamentul functiei **strlcpy** (man strlcpy).
- Ea trebuie sa aiba urmatorul prototip:

```
unsigned int ft_strlcpy(char *dest, char *src, unsigned int size);
```

Capitolul XXIII

Exercitiu 20 : ft_putnbr_base

	Exercitiu: 20
ft_putnbr_base	
Director de lucru: ex20/	
Fisier(e) de iesire: ft_putnbr_base.c	
Functii autorizate: ft_putchar	
Observatii: n/a	


- Scrieti o functie care afiseaza un numar intr-o baza data.
- Acest numar este dat sub forma unui `int` si baza sub forma `sir de caractere`.
- Baza contine toate simbolurile utilizabile pentru afisarea numarului:
 - 0123456789 este baza curenta utilizata pentru reprezentarea numerelor zecimale;
 - 01 este o baza binara;
 - 0123456789ABCDEF este o baza hexazecimala;
 - poneyvif este o baza octala.
- Functia trebuie sa tina seama si de numerele negative.
- Daca un parametru contine o eroare functia nu afiseaza nimic. O eroare poate fi:
 - baza este vida sau este de marimea 1;
 - baza contine de doua ori aceleasi caracter;
 - baza contine caracterele + sau - ;
 - etc.

- Ea trebuie sa aiba urmatorul prototip:

```
void ft_putnbr_base(int nbr, char *base);
```

Capitolul XXIV

Exercitiu 21 : ft_atoi_base


	Exercitiu: 21
	ft_atoi_base
Director de lucru: ex21/	
Fisier(e) de iesire: ft_atoi_base.c	
Functii autorizate: Niciuna	
Observatii: n/a	

- Scrieti o functie care returneaza un numar. Acest numar este dat sub forma unui sir de caractere.
- Sirul de caractere defineste numărul într-o anumită bază, data ca al doilea parametru.
- Functia trebuie sa tina seama si de numere negative.
- Functia trateaza semnele ca in `man atoi`.
- Daca un parametru contine o eroare functia returneaza 0. O eroare poate fi:
 - sirul de caractere este vid;
 - baza este vida sau egala cu 1;
 - sirul contine caractere care nu sunt din baza data si nu sunt + sau - ;
 - baza contine de doua ori acelasi caracter;
 - baza contine caracterele + sau - ;
 - etc.
- Ea trebuie sa aiba urmatorul prototip:

```
int      ft_atoi_base(char *str, char *base);
```

Capitolul XXV

Exercitiu 22 : ft_putstr_non_printable

	Exercitiu: 22
ft_putstr_with_non_printable	
Director de lucru: ex22/	
Fisier(e) de iesire: ft_putstr_non_printable.c	
Functii autorizate: ft_putchar	
Observatii: n/a	

- Scrieti o functie care afiseaza un sir de caractere. Daca acest sir contine caractere netiparibile, ele trebuie afisate in format hexazecimal (cu minuscule) si precedate de caracterul "backslash".
- De exemplu, cu acest parametru:

```
Salut \nce mai faci?
```

- Functia trebuie sa afiseze:


```
Salut \0ce mai faci?
```

- Ea trebuie sa aiba prototipul urmator:

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

Capitolul XXVI

Exercitiu 23 : ft_print_memory

	Exercitiu: 23
ft_print_memory	
Director de lucru: <i>ex23/</i>	
Fisier(e) de iesire: <code>ft_print_memory.c</code>	
Functii autorizate: <code>ft_putchar</code>	
Observatii: n/a	

- Scrieti o functie care afiseaza o zona de memorie pe ecran.
- Afisarea zonei de memorie se va face pe trei coloane:
 - Adresa in hexazecimal a primului caracter al liniei;
 - Continutul in hexazecimal;
 - Continutul in caractere tiparibile.
- Daca un caracter este netiparibil el va fi inlocuit cu un punct.
- Fiecare linie trebuie sa afiseze saispnezeze caractere.
- Daca `size` este egal cu 0, nu se va afisa nimic.

- Exemplu:

```
guilla_i@seattle $> ./ft_print_memory
00000000: 5361 6c75 7420 6c65 7320 616d 696e 6368 Salut les aminch
00000010: 6573 2063 2765 7374 2063 6f6f 6c20 7368 es c'est cool sh
00000020: 6f77 206d 656d 206f 6e20 6661 6974 2064 ow mem on fait d
00000030: 6520 7472 7563 2074 6572 7269 626c 6500 e truc terrible.
00000040: 2e00 0102 0304 0506 0708 090e 0f1b 7f .....
guilla_i@seattle $> ./ft_print_memory | cat -te
00000000: 5361 6c75 7420 6c65 7320 616d 696e 6368 Salut les aminch$
00000010: 6573 2063 2765 7374 2063 6f6f 6c20 7368 es c'est cool sh$
00000020: 6f77 206d 656d 206f 6e20 6661 6974 2064 ow mem on fait d$
00000030: 6520 7472 7563 2074 6572 7269 626c 6500 e truc terrible.$
00000040: 2e00 0102 0304 0506 0708 090e 0f1b 7f .....$
guilla_i@seattle $>
```

- Ea trebuie sa aiba urmatorul prototip:

```
void      *ft_print_memory(void *addr, unsigned int size);
```

- Ea trebuie sa returneze addr.