



Bazén C

C 07

Zaměstnanci 42 piscine@42.fr

Shrnutí: Tento dokument je předmětem modulu C 07 skupiny C 42.

Obsah

I	Instrukce	2
II	Preamble	4
III	Cvičení 00: ft_strdup	5
IV	Cvičení 01 : rozsah stop	6
V	Cvičení 02: ft_ultimate_range	7
VI	Cvičení 03: ft_strjoin	8
VII	Cvičení 04: ft_convert_base	9
VIII	Cvičení 05: ft_split	10

Kapitola I

Instrukce

- Pouze tato stránka bude sloužit jako reference: nespolehejte na fámy z chodby.
- Před odesláním cvičení si předmět znovu dobře přečtěte. Subjekt může kdykoli měnič.
- Věnujte pozornost právům vašich souborů a adresářů.
- Pro všechna cvičení musíte dodržet postup vykreslování.
- Vaše cvičení opraví vaši kolegové z bazénu.
- Kromě vašich spolubojovníků vás opraví program zvaný Mou linkette.
- La Moulinette je velmi přísná ve svém bodování. Je plně automatizovaný. Je nemožné s ní diskutovat o její známce. Buďte bezúhonní, abyste se vyhnuli překvapení.
- La Moulinette není příliš otevřená. Nesnaží se porozumět kódu, který nesplňuje standard. La Moulinette používá program norminette ke kontrole standardu vašich souborů. Pochopte tím, že je hloupé vykreslovat kód, který neprojde normou.
- Cviky jsou velmi přesně seřazeny od nejjednodušších po nejsložitější. V žádném případě nebudeme věnovat pozornost nebo brát v úvahu komplexní cvičení, pokud jednodušší cvičení není dokonale úspěšné.
- Použití zakázané funkce je případem podvádění. Veškeré podvádění je bezpečné se skóre -42.
- Funkci main() budete muset vrátit pouze v případě, že vás požádáme o program. _____
- La Moulinette se kompiluje s příznaky -Wall -Wextra -Werror a používá gcc.
- Pokud se váš program nezkompiluje, dostanete 0.
- Ve svém adresáři byste neměli ponechat žádný jiný soubor než ty explicitní ment specifikovaný ve výkazech cvičení.
- Máte dotaz? Zeptejte se svého souseda vpravo. Jinak zkuste s

váš soused vlevo.

- Vaše referenční příručka se nazývá Google /man/Internet/....
- Nezapomeňte diskutovat na fóru o plaveckém bazénu vašeho Intra, stejně jako o volném čase
Plavecký bazén !
- Přečtěte si pozorně příklady. Mohou vyžadovat věci, které ne
není v předmětu uvedeno jinak...
- Myslet si. Prosím, od Odina! Název dýmky.



Pro tento den musí být norminette spuštěna s příznakem -R
CheckForbiddenSourceHeader. Využijte to i mlynek.

Kapitola II


Preamble

Zde je seznam příšer, které lze nalézt ve slavném Naheul beuk Dungeonu:

- Všechny druhy nemrtvých; - Obří pavouci; - Kosatky;
- Goblini; - Trollové v podzemí; - Čarodějové; -
- Prokletí válečníci; - mutantní krysy; - Láhev oleje; -
- Toaletní papír; - Dvě houby; - Ravioli.

Kapitola III

Cvičení 00: ft_strdup


	Cvičení: 00
	ft_strdup
Renderová složka: ex00/	
Soubory k vykreslení: ft_strdup.c	
Povolené funkce: malloc	

- Stejným způsobem reprodukuje činnost funkce strdup (man strdup).
- Musí být prototypován následovně:

```
char *ft_strdup(char *src);
```

Kapitola IV

Cvičení 01: rozsah stop

	Cvičení: 01
ft_range	
Renderová složka: ex01/	
Soubory k vykreslení: ft_range.c	
Povolené funkce: malloc	


- Napište funkci rozsah_ft, která vrátí pole ints. Toto pole int bude obsahovat všechny hodnoty mezi min a max.
- Min v ceně - max bez.
- Musí být prototypován následovně:

```
int *ft_range(int min, int max);
```

- Pokud je minimální hodnota větší nebo rovna maximální hodnotě, bude nulový ukazatel vrátit se.

Kapitola V

Cvičení 02: ft_ultimate_range

	Cvičení: 02
ft_ultimate_range	
Složka vykreslení: ex02/	
Soubory k vykreslení: ft_ultimate_range.c	
Povolené funkce: malloc	


- Napište funkci `ft_ultimate_range`, která alokuje a přiřadí pole `ints`. Toto pole `int` bude obsahovat všechny hodnoty mezi `min` a `max`.
- `Min` v ceně - `max` bez.
- Musí být prototypován následovně:

```
int ft_ultimate_range(int **rozsah, int min, int max);
```

- Bude vrácena velikost rozsahu (nebo -1 v případě problému).
- Pokud je minimální hodnota větší nebo rovna maximální hodnotě, bude rozsah ukazovat na `NULL` a vrátíme 0.

Kapitola VI

Cvičení 03: ft_strjoin


	Cvičení: 03
	ft_strjoin
Renderová složka: ex03/	
Soubory k vykreslení: ft_strjoin.c	
Povolené funkce: malloc	

- Napište funkci, která zřetězí všechny špičaté řetězce znaků pomocí str, které je oddělí pomocí sep.
- size představuje velikost str.
- Pokud je velikost 0, musíte vrátit prázdný řetězec znaků, který můžete uvolnit().
- Musí být prototypován následovně:

```
char *ft_strjoin(int size, char **strs, char *sep);
```

Kapitola VII

Cvičení 04: ft_convert_base


	Cvičení: 04
ft_convert_base	
Renderová složka: ex04/	
Soubory k vykreslení: ft_convert_base.c, ft_convert_base2.c	
Povolené funkce: malloc, free	

- Napište funkci, která vrátí výsledek převodu řetězce nbr vyjádřeného v základu base_from v základu base_to.
- nbr, base_from, base_to nemusí být nutně modifikovatelné.
- nbr se bude řídit stejnými pravidly jako ft_atoi_base. Věnujte pozornost '+', '-' a mezery.
- Číslo reprezentované nbr se vejde do int.
- Pokud je základ nesprávný, funkce vrátí hodnotu NULL.
- Vracené číslo by mělo být v případě potřeby opatřeno předponou pouze jedním '-', bez mezer nebo '+'.
'+'.
- Musí být prototypován následovně:

```
char *ft_convert_base(char *    char *base_from, char *base_to);
```

Kapitola VIII

Cvičení 05: ft_split

	Cvičení: 05
ft_split	
Renderová složka: ex05/	
Soubory k vykreslení: ft_split.c	
Povolené funkce: malloc	

- Napište funkci, která rozdělí jeden řetězec na druhý řetězec znaků.
- Každý znak v řetězci znakové sady musí být použit jako oddělovač.
- Funkce vrací pole, kde každý prvek tohoto pole obsahuje adresu znakového řetězce mezi dvěma oddělovači. Poslední prvek pole musí být roven 0, aby označoval konec pole.
- V poli nesmí být žádné prázdné řetězce. Vyvodit závěry které jsou vyžadovány.
- Řetězec, který bude přenášen, nebude možné upravit.
- Musí být prototypován následovně:

```
char **ft_split(char *str, char *charset);
```