

{Epitech}

Introduction au C

VERMERSCH Adrien

7 décembre 2021



Introduction

1. Hello World !
2. Hello World ! V.2
3. Découverte des “string”
4. Réception des paramètres
5. Les “string” variables (malloc)
6. Afficher des numéros
7. Recoder printf

Astuce : Lisez les man, puis demandez de l’aide.

N'utilisez pas internet.



Hello World !

Objectif : Pouvoir afficher "Hello World !" dans son terminal.

```
int main(void) ;
```

Fonction.s autorisée.s :

- printf

Astuce : n'hésite pas à faire un “man 3 printf” dans le terminal pour savoir comment utiliser cette fonction.



Hello World ! (v2)

Objectif : Pouvoir afficher "Hello World !" dans son terminal

```
int main(void) ;
```

Fonction.s autorisée.s :

- write

Astuce : n'hésite pas à faire un “man 2 write” dans le terminal pour savoir comment utiliser cette fonction.



Découverte des “string”

Objectif : Pouvoir afficher une string dans son terminal

```
int main(void) ;
```

Fonction.s autorisée.s :

- write

Astuce : n'hésite pas à faire un “man 2 write” dans le terminal pour savoir comment utiliser cette fonction.



Réception des paramètres

Objectif : Pouvoir afficher à l'aide de printf une string qui nous a été envoyée par l'utilisateur

```
int main(int argc, char **argv);
```

Fonction.s autorisée.s :

- printf

Astuce : n'hésite pas à faire un “man printf” dans le terminal pour savoir comment utiliser cette fonction.

Les “string” variables (malloc)

Objectif : Pouvoir dupliquer la moitié de la string donnée par l'utilisateur, et l'afficher.

```
int main(int argc, char **argv);  
  
char *my_str_semi_dup(char *str);  
  
void my_putstr(char *str);
```

Fonction.s autorisée.s :

- write
- malloc

Astuce : n'hésite pas à faire un “man malloc” / “man 2 write” dans le terminal pour savoir comment utiliser cette fonction. Une string finit toujours par un ‘\0’.



Afficher des numéros

Objectif : écrire une fonction qui va afficher les nombres qui lui sont donnés en paramètre. Elle doit pouvoir supporter toutes les valeurs possibles d'un int.

```
int my_putnbr(int nb);
```

Fonction.s autorisée.s :

- write

Astuce : n'hésite pas à faire un “man 2 write” dans le terminal pour savoir comment utiliser cette fonction. Une fonction my_putchar qui affiche un caractère unique peut être utile.

■ ■ ■

Très bien, vous êtes chauds !

On va corser tout ça et passer au recodage d'une partie de la fonction printf. :)



MyPrintf

Objectif : recoder printf

```
int my_printf(char *str, ...);
```

Fonction.s autorisée.s :

- malloc
- write

Astuce : n'hésite pas à faire un “man 3 printf” / “man 3 stdarg” dans le terminal pour savoir comment fonctionnent ces fonctions. Il est conseillé de tester les fonctions une par une, puis, assembler le tout dans my_printf.

Pour ce my_printf, il vous faut refaire le flag **%c**, **%s**, **%d**, **%i**, **%X** et un petit custom, le **%b**, qui devra afficher le binaire d'un nombre.

Avant de commencer, demande à une personne de t'expliquer, parce qu'on a des choses à raconter.

1. il faudra probablement utiliser des listes variadiques
2. il faudra pouvoir calculer la longueur de la string passer en argument
3. il faut être capable de détecter les flags pour faire l'action qui lui correspond



Si tu as tout fini, essaye d'améliorer ton code, ou va plus loin en faisant tous les flags de printf.

Bonne chance !