

```
// workflow C (exemple avec c00 ex00 (ft_putchar.c)) :
```

```
    // regardez les vidéos de c00  
    // ouvrez fr.norm.pdf  
    // ouvrez c00-intro-fr.subject.pdf  
    // ouvrez c00-ex00-fr.subject.pdf  
    // ouvrez le terminal
```

```
cd Piscine  
cd c00  
cd ex00
```

```
    // créez votre fichier ft_putchar.c avec votre fonction :
```

```
vim ft_putchar.c  
<F1>  
<i>  
#include <unistd.h>  
  
void    ft_putchar(char c)  
{  
    write(1, &c, 1);  
}  
<escape>  
:wq  
<enter>
```

```
    // exécutez ces commandes dans une autre fenêtre du terminal :
```

```
    // créez votre fichier ft_putchar.h avec le prototype de votre fonction :
```

```
vim ft_putchar.h  
<F1>  
<i>  
void    ft_putchar(char c);  
<escape>  
:wq  
<enter>
```

// créez votre fichier main.c pour tester votre fonction :

vim main.c

<F1>

<i>

#include "ft\_putchar.h"

int main(void)

{

ft\_putchar('a');

}

<escape>

:wq

<enter>

// ou

vim main.c

<F1>

<i>

#include "ft\_putchar.h"

int main(int argc, char \*argv[])

{

if (argc > 1)

{

ft\_putchar(argv[1][0]);

}

}

<escape>

:wq

<enter>

// compilez votre fichier main.c avec votre fichier ft\_putchar.c :

gcc -Wall -Wextra -Werror ft\_putchar.c main.c

// exécutez votre main :

./a.out

```
        // ou
./a.out a

        // exécutez la norminette :
norminette -R CheckForbiddenSourceHeader ft_putchar.c

        // exécutez la mini moulinette :
cd ..
mini C00

        // exécutez le semi-auto-correct-42 :
cd c00
mkdir eval
mkdir eval/ex00
cd ex00
cp ft_putchar.* ../eval/ex00
cd ..
../../semi-auto-correct-42/correct.sh
<CTRL><C>

        // testez votre code sur pythontutor C :
https://pythontutor.com/c.html#mode=edit

        // comparez votre code avec les piscineux :
https://github.com/suker/42-Piscine
https://github.com/pasqualerossi/42-Piscine/

        // poussez c00 ex00
git status
git add .
git status
git commit -m "c00ex00"
git push
```