

Tests de code :

Test de nombre très grands :

```
#include "main.h"

#include <limits.h>
#include <stdio.h>
#include "main.h"

/**
 * main - Entry point
 *
 * Return: Always 0
 */

int main() {
    long bigNum = 2147483648L;
    unsigned long veryBigNum;
    veryBigNum = 100000000000000UL;
    printf("Grand nombre : %ld\n", bigNum);
    printf("Nombre très grand : %lu\n", veryBigNum);
    return 0;
}
```

Résultat :

```
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ gcc -Wall -Werror -Wextra -pedantic -std=gnu89 -Wno-format *.c
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ ./a.out
Grand nombre : 2147483648
Nombre très grand : 100000000000000
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ |
```

Test du main donné :

```

* Return: Always 0
*/

#include <limits.h>
#include <stdio.h>
#include "main.h"

/**
 * main - Entry point
 *
 * Return: Always 0
 */
int main(void)
{
    int len;
    int len2;
    unsigned int ui;
    void *addr;

    len = _printf("Let's try to printf a simple sentence.\n");
    len2 = printf("Let's try to printf a simple sentence.\n");
    ui = (unsigned int)INT_MAX + 1024;
    addr = (void *)0x7ffe637541f0;
    _printf("Length:[%d, %i]\n", len, len);
    printf("Length:[%d, %i]\n", len2, len2);
    _printf("Negative:[%d]\n", -762534);
    printf("Negative:[%d]\n", -762534);
    _printf("Unsigned:[%u]\n", ui);
    printf("Unsigned:[%u]\n", ui);
    _printf("Unsigned octal:[%o]\n", ui);
    printf("Unsigned octal:[%o]\n", ui);
    _printf("Unsigned hexadecimal:[%x, %X]\n", ui, ui);
    printf("Unsigned hexadecimal:[%x, %X]\n", ui, ui);
    _printf("Character:[%c]\n", 'H');
    printf("Character:[%c]\n", 'H');
    _printf("String:[%s]\n", "I am a string !");
    printf("String:[%s]\n", "I am a string !");
    _printf("Address:[%p]\n", addr);
    printf("Address:[%p]\n", addr);
    len = _printf("Percent:[%%]\n");
    len2 = printf("Percent:[%%]\n");
    _printf("Len:[%d]\n", len);
    printf("Len:[%d]\n", len2);
    _printf("Unknown:[%r]\n");
    printf("Unknown:[%r]\n");
    return (0);
}

```

Résultat :

```
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ gcc -Wall  
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ ./a.out  
Let's try to printf a simple sentence.  
Let's try to printf a simple sentence.  
Length:[39, 39]  
Length:[39, 39]  
Negative:[-762534]  
Negative:[-762534]  
Unsigned:[%u]  
Unsigned:[2147484671]  
Unsigned octal:[%o]  
Unsigned octal:[20000001777]  
Unsigned hexadecimal:[%x, %X]  
Unsigned hexadecimal:[800003fff, 800003FF]  
Character:[H]  
Character:[H]  
String:[I am a string !]  
String:[I am a string !]  
Address:[%p]  
Address:[0x7ffe637541f0]  
Percent:[%]  
Percent:[%]  
Len:[12]  
Len:[12]  
Unknown:[%r]  
Unknown:[%r]  
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ |
```

Test simple d'âge :

```
#include "main.h"

#include <limits.h>
#include <stdio.h>
#include "main.h"

/**
 * main - Entry point
 *
 * Return: Always 0
 */

int main() {
    int age = 30;
    _printf("L'âge est: %d\n", age);
    return 0;
}
```

Résultat :

```
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ ./a.out
L'âge est: 30
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ |
```

Test d'une string :

```
#include "main.h"

#include <limits.h>
#include <stdio.h>
#include "main.h"

/**
 * main - Entry point
 *
 * Return: Always 0
 */

int main() {
    _printf("Bonjour, le monde!\n");
    return 0;
}
```

Résultat :

```
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ gcc -Wall -Werror -Wextra -pedantic -std=gnu89 -Wno-format *.c
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ ./a.out
Bonjour, le monde!
josephine@LAPTOP-E3L90469:~/holbertonschool-printf$ |
```