

May 14, 21 18:39

compteur.c

Page 1/1

indentation

```
#include <stdio.h>

int compteur(){
    static int cpt = -1;
    cpt ++;
    return cpt;
}

int main(){
    int i, res;
    for(i = 0; i < 10; i++){
        res = compteur();
        printf("%d\n", res);
    }
    return 0;
}
```

B⁺

May 14, 21 18:39

mstrdup.c

Page 1/1

```

#include <stdlib.h>
#include <stdio.h>

char *mstrdup(const char *str) {
    int i = 0;
    int cpt = 0;
    while (str[i] != '\0') {
        cpt++;
        i++;
    }
    // inutile ici .

    char * res = malloc((cpt + 1) * sizeof(char));

    if (res == NULL) {
        return NULL;
    }

    for (i=0; i<cpt; i++) {
        res[i] = str[i];
    }

    res[i] = '\0';

    return res;
}

int main(int argc, char *argv[]){
    int i;
    char *result;
    if (argc > 1) {
        for (i=0; i<argc; i++){
            result = mstrdup(argv[i]);
            printf("%s\n", result);
            free(result);
        }
    }
    exit(EXIT_SUCCESS);
}

/*
si il y a un problème lors de l'allocation de mémoire avec malloc, la fonction r
envoie NULL
*/

```

May 14, 21 18:39

nom_temp.c

Page 1/1

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```
char * nom_temp(){
    static int cpt = 0;
    static char res[] = "FILE";
    res[4] = (cpt++) + '0';
    return res;
}
```

res[4] = ...; écrase le '0' final. Solutions:

- static char res[5] = "FILE";
- static char res[] = "FILE0";

```
char *nom_temp2(){
    char *res = malloc(6 * sizeof(char));
    static int cpt = 0;

    if (res == NULL) {return NULL;}

    snprintf(res, 6, "FILE%d", cpt++);
    return res;
}
```

|| N'imprimera que le premier chiffre de cpt dans res, donc va générer plusieurs fois le même nom successivement:

- cpt == 10 => FILE1
- cpt == 11 => FILE1
- ⋮

```
void test_nom_temp() {
    char *file1, *file2, *file3;
    printf("\nTest 1\n");

    file1 = nom_temp();
    printf("Fichier 1: %s\n", file1);

    file2 = nom_temp();
    printf("Fichier 2: %s\n", file2);
    file3 = nom_temp();
    printf("Fichier 3: %s\n", file3);
}
```

```
void test_nom_temp2(){
    char *file1, *file2, *file3, *file4;
    printf("\nTest 2\n");

    file1 = nom_temp2();
    printf("Fichier 1: %s\n", file1);
    free(file1);

    file2 = nom_temp2();
    printf("Fichier 2: %s\n", file2);
    free(file2);

    file3 = nom_temp2();
    printf("Fichier 3: %s\n", file3);
    free(file3);

    file4 = nom_temp2();
    printf("Fichier 4: %s\n", file4);
    free(file4);
}
```

```
int main(){
    test_nom_temp();
    test_nom_temp2();
    return 0;
}
```

Il manque les questions 2 et 4.