

A⁺

May 14, 21 18:42

compteur.c

Page 1/1

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

/* Question 1 */
int compteur(){
    static int cpt = 0;
    return cpt++;
}

/*Question 2 */
int main(){
    int i;

    printf("Compteur:\n");
    for (i=0; i<10; i++){
        printf("%d\n",compteur());
    }
    return 0;
}

/*
    On peut aussi utiliser le pointeur
int compteur(int *ptr){
    return (*ptr)++;
}
*/
```

1✓

1✓

~~X~~ nan

May 14, 21 18:42

dupliquer.c

Page 1/1

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

/* Question 1
On a un problème lorsque la mémoire est
insuffisante pour allouer une chaîne en double
*/
char *mstrdup(const char *str){
    size_t len = strlen(str) + 1;
    char *new = malloc(len);
    if (new == NULL)
        return NULL;
    return memcpy(new, str, len);
}

/*Question 2 */
int main(int argc, char *argv[]){
    int i;
    char * copy = NULL;

    for(i=0; i< argc; i++){
        copy = mstrdup(argv[i]);
        printf("Address de copie : %p\n", (void *)copy);
        printf("Address de argv[%d]: %p\n", i, (void *)argv[i]);
        printf("%s ", copy);
        puts("\n");
        free(copy);
    }
    return 0;
}
```

✓

✓

May 14, 21 18:42

nom_temp.c

Page 1/2

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 6
```

```
/*Question 1, 3*/
char *nom_temp1(){
    static int ctr = 0;
    static char tab[] = "fileN";
    tab[4] = (ctr++) + '0';
    return tab;
}
```

```
char *nom_temp2(){
    char *new = malloc(MAX * sizeof(char));
    static int inc = 0;
    if (new == NULL)
        return NULL;
    snprintf(new, MAX, "file%d", inc++);
    return new;
}
```

```
/*
Question 2
Type static et dynamique
*/
```

```
/*
Question 4
l'impact est que nous aurons la même valeur pour
chaque test (la valeur de la dernière appel),
c'est parce que on a utilisé des variables statiques
et chaque appel remplacera la valeur précédente
*/
```

```
void test_nom_temp1() {
    char *file1, *file2, *file3, *file4;
    puts("Type Static : \n");

    file1 = nom_temp1();
    printf("Fichier 1: %s\n", file1);
    file2 = nom_temp1();
    printf("Fichier 2: %s\n", file2);
    file3 = nom_temp1();
    printf("Fichier 3: %s\n", file3);
    file4 = nom_temp1();
    printf("Fichier 4: %s\n\n", file4);
}
```

```
/*Question 5*/
void test_nom_temp2() {
    char *file1, *file2, *file3, *file4;
    puts("Type Dynamique : \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, pas char.

→ imprime uniquement le premier chiffre de inc, donc générera plusieurs fois le même nom successivement :

inc == 10 ⇒ file 1
inc == 11 ⇒ file 1
:

de.

May 14, 21 18:42

nom_temp.c

Page 2/2

```
int main(void) {
    test_nom_temp1();
    test_nom_temp2();
    return 0;
}
```