

Programmation C Exemple de correction TP 2

Voici un des SEULS corrigés de l'année (probablement le seul). Ce dernier est destiné à la fois aux débutants mais aussi à ceux qui sont plus à l'aise. Ce corrigé se veut un peu comme un exemple de proposition propre que l'on peut faire pour un TP au niveau bac+3.

Code sources :

fichier exo1.c :

```
1 #include <stdio.h>
2
3 /* The classic C Hello World program. */
4 int main(int argc, char* argv[]){
5     printf("Hello World!\n");
6     return 0;
7 }
```

fichier exo2.c :

```
1 #include <stdio.h>
2
3 /* Print the sum of two integers given by the user in the terminal. */
4 int main(int argc, char* argv[]){
5     int a, b, sum;
6
7     printf("Give me two integers separated by a space:\n");
8     scanf("%d%d", &a, &b);
9     sum = a + b;
10    printf("The sum of %d and %d is %d\n", a, b, sum);
11    return 0;
12 }
```

fichier exo2_bis.c :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Print the sum of two integers given by the
5    user at call of the program. */
6 int main(int argc, char* argv[]){
7     int a, b, sum;
8
9     if (argc != 3){
10         fprintf(stderr, "Error: this program needs"
11                    "\n exactly two integers as arguments\n");
```

```

12     return 1;
13 }
14 a = atoi(argv[1]);
15 b = atoi(argv[2]);
16 sum = a + b;
17 printf("The sum of %d and %d is %d\n", a, b, sum);
18 return 0;
19 }

```

fichier exo3.c :

```

1  #include <stdio.h>
2
3  /* Display n n-1 ... 2 1 1 2 ... n-1 n using for instructions. */
4  void number_imperative(int n){
5      int i;
6
7      for (i=n ; i>0 ; i--){
8          printf("%d ", i);
9      }
10     for (i=1 ; i<=n ; i++){
11         printf("%d ", i);
12     }
13     printf("\n");
14 }
15
16 /* Display n n-1 ... 2 1 1 2 ... n-1 n using the recursion. */
17 void number_recursive(int n){
18     if (n >= 1){
19         printf("%d ", n);
20         number_recursive(n-1);
21         printf("%d ", n);
22     }
23 }
24
25 int main(int argc, char* argv[]){
26     int i = 10;
27
28     printf("Imperative version\n");
29     number_imperative(i);
30     printf("Recursive version\n");
31     number_recursive(i);
32     printf("\n");
33     return 0;
34 }

```

fichier exo4.c :

```

1  #include <stdio.h>
2
3  /* Display just the integer n. */

```

```

4 void print_var(int n){
5     printf("Value of the variable: %d\n", n);
6 }
7
8 /* Display the address of the pointer and the pointed value. */
9 void print_pointer(int* p){
10     printf("Pointer address: %p and Pointed value: %d\n", p, *p);
11 }
12
13 /* Set the pointed value *p of the pointer p to n. */
14 void set_pointer(int* p, int n){
15     *p = n;
16 }
17
18 int main(int argc, char* argv[]){
19     int a;
20     int* p=&a;
21
22     print_var(a);
23     a = 53;
24     print_var(a);
25     print_pointer(p);
26     set_pointer(p, 42);
27     /* set_pointer affect the local variable 'a'
28        without taking it as an argument. */
29     print_pointer(p);
30     print_var(a);
31
32     return 0;
33 }

```

fichier exo5.c :

```

1 #include <stdio.h>
2
3 /* This program open its own source file
4    and display it on the standard output. */
5 int main(int argc, char* argv[]){
6     FILE* fichier;
7     char c;
8
9     fichier = fopen("exo5.c", "r");
10    while ((c = fgetc(fichier)) != EOF){
11        putchar(c);
12    }
13    return 0;
14 }

```

Makefile :

```
# Makefile TP2 - Prog C IR1/IG1
```

```

all:
    make exo1
    make exo2
    make exo3
    make exo4
    make exo5

exo1: exo1.c
    gcc -o HelloWorld exo1.c -Wall -ansi

exo2: exo2.c exo2_bis.c
    gcc -o Sum exo2.c -Wall -ansi
    gcc -o Sum_bis exo2_bis.c -Wall -ansi

exo3: exo3.c
    gcc -o Number exo3.c -Wall -ansi

exo4: exo4.c
    gcc -o Pointer exo4.c -Wall -ansi

exo5: exo5.c
    gcc -o Replicant exo5.c -Wall -ansi

clean:
    rm HelloWorld
    rm Sum
    rm Sum_bis
    rm Number
    rm Pointer
    rm Replicant
    rm *~

```

Effets produits en machine :

```

nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ls
exo1.c exo2_bis.c exo2.c exo3.c exo4.c exo5.c Makefile
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ make
make exo1
make[1]: entrant dans le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
gcc -o HelloWorld exo1.c -Wall -ansi
make[1]: quittant le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
make exo2
make[1]: entrant dans le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
gcc -o Sum exo2.c -Wall -ansi
gcc -o Sum_bis exo2_bis.c -Wall -ansi
make[1]: quittant le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
make exo3
make[1]: entrant dans le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
gcc -o Number exo3.c -Wall -ansi
make[1]: quittant le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
make exo4
make[1]: entrant dans le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
gcc -o Pointer exo4.c -Wall -ansi
make[1]: quittant le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
make exo5
make[1]: entrant dans le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
gcc -o Replicant exo5.c -Wall -ansi
make[1]: quittant le répertoire « /home/nborie/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige »
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ls
exo1.c exo2_bis.c exo2.c exo3.c exo4.c exo5.c

```

```

HelloWorld Makefile Number Pointer Replicant Sum Sum_bis
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./HelloWorld
Hello World!
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Sum
Give me two integers separated by a space:
234 4532
The sum of 234 and 4532 is 4766
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Sum_bis 23
Error: this program needs exactly two integers as arguments
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Sum_bis 23 45
The sum of 23 and 45 is 68
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Number
Imperative version
10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
Recursive version
10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Pointer
Value of the variable : 32767
Value of the variable : 53
Pointer address : 0x7fff6ddc84d4 and Pointed value : 53
Pointer address : 0x7fff6ddc84d4 and Pointed value : 42
Value of the variable : 42
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$ ./Replicant
#include <stdio.h>

/* This program open its own source file
   and display it on the standard output. */
int main(int argc, char* argv[]){
    FILE* fichier;
    char c;

    fichier = fopen("exo5.c", "r");
    while ((c = fgetc(fichier)) != EOF){
        putchar(c);
    }
    return 0;
}
nborie@perceval:~/Enseignement/Marne_2014_2015/C_IR1/tp/tp2/corrige$

```

Si ce TP avait été à rendre, il aurait fallu alors produire une archive tar.gz des fichiers sources (les .c) avec le fichier Makefile. Le correcteur n'a plus qu'à taper make pour compiler tous les exercices.

Ne pas oublier de faire make clean avant un rendu, le correcteur se moque de vos exécutables, il veut les sources!!!

Q: Ouais mais monsieur, il est super long votre corrigé !!! On n'a pas le temps de faire tout ça en deux heures de TP !

R: Mais si! On va apprendre à programmer vite et bien. Les bonnes habitudes sont dures à prendre mais en les pratiquant au plus vite et systématiquement, ce sont vos doigts qui vont les mémoriser!