

Fiche n° 3 de TP

Fonctions

Objectifs : écriture et utilisation de fonctions.

Prérequis : syntaxe des fonctions ; passage des paramètres.

Travail minimum : exercices 1 à 3.

Exercice 1

Tapez sans y apporter la moindre modification le programme suivant dans le fichier `eval-poly.c`.

```
eval-poly.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 double eval_poly(double x);
5
6 int main(void) {
7     printf("Entrez une valeur réelle : ");
8     double v;
9     scanf("%lf", &v);
10
11     double r = eval_poly(v);
12     printf("La valeur du polynome pour %lf est %lf.\n", v, r);
13
14     return EXIT_SUCCESS;
15 }
16
17 double eval_poly(double x) {
18     double resultat;
19
20     resultat = 3 * x * x - 4 * x + 1;
21
22     return resultat;
23 }
```

eval-poly.c

Exécutez-le avec les valeurs 1 puis 2. Que fait la fonction `eval_poly` ?

Exercice 2

- 1) Écrivez une fonction qui prend en paramètre deux réels et qui renvoie leur moyenne.
- 2) Écrivez ensuite un programme qui utilise cette fonction pour déterminer et afficher la moyenne de quatre valeurs réelles données en entrée.

Exercice 3

- 1) Écrivez une fonction qui prend en paramètre trois entiers h , m et s représentant une valeur horaire exprimée sous la forme « heure minute seconde » et qui retourne cette valeur horaire exprimée en secondes. Vous pourrez vous inspirer du programme que vous avez écrit pour l'exercice 2 de la fiche de TP n° 2.
- 2) En déduire un programme `time_to_sec` qui lit sur l'entrée une valeur horaire exprimée sous la forme « heure:minute:seconde » et qui affiche cette valeur exprimée en secondes :

```
$ ./time_to_sec
12:34:56
45296
```

- 3) Que produit la commande suivante? Expliquez.

```
echo "12:34:56" | ./time_to_sec > conv_sec.txt
```

- 4) Écrivez une fonction qui convertit une durée exprimée en seconde en une valeur horaire exprimée sous la forme « heure minute seconde ». Indice : cette fonction, qui ne retournera rien, pourra accepter quatre paramètres, un passé par valeur, les trois autres par adresse.
- 5) En déduire un programme `sec_to_time` qui lit sur l'entrée une durée exprimée en seconde et qui affiche cette valeur sous la forme « heure:minute:seconde » :

```
$ ./sec_to_time  
45296  
12:34:56
```

- 6) Vérifiez que le second programme est bien le pendant du premier en faisant lire au premier la sortie du second et en vérifiant que la valeur affichée est bien la même que celle que vous avez rentrée.