

0.1 Introduction au C

Le langage C a vu jour dans les années 70 et a pour fondateurs : **D.Ritchie & B.W.Kernighan**. Leurs objectifs avec ce nouveau langage étaient l'obtention d'un nouveau système d'exploitation de type UNIX portable sur différentes plateformes. Ils se sont fortement inspirés des langages **B** et **BCPL** ; d'où ce nom : **C**.

Ces quelques pages sur le langage C n'ont pas la prétention de vous apprendre à coder en C mais simplement de vous donner une idée de ce langage. Pourquoi s'intéresser au C ? Outre qu'il est possible durant votre vie que vous le rencontriez pour une raison ou une autre, que ce soit un langage fabuleux par toute la puissance qu'il renferme, c'est avant tout le langage sur lequel repose tout GNU/Linux.

Je vous propose pour cette introduction de regarder un petit programme d'abord en algorithme puis le code C correspondant. Le programme fera les choses suivantes : il demandera à l'utilisateur de rentrer un à un des relevés de températures (nom de la ville, température maximum, température minimum). Il calculera ensuite l'écart maximum de température enregistré sur l'ensemble de ces relevés et son arrondi, et donnera le nombre de ville ayant enregistrées la température maximale. Il affichera ces résultats à l'utilisateur.

```
/*Déclaration des constantes : */
zero=-273.15
/*Ecriture des différentes fonctions appelées par la fonction
  principale : */
fonction arr(a)
/*La fonction arr(a) donne l'arrondi du réel a.
Il est par exemple de 5 pour 5,1; de 6 pour 5,6 et enfin
  de 6 pour la valeur 5,5 */
variables globales :
arr() :entier
a :réel
variables locales :
Début
arr <- E(a) +0.5; /* où E() est la fonction partie entière
  */
Fin
fonction entree ()
/*La fonction entree permet de connaître le choix de l'utilisateur
  sur sa volonté de continuer ou non.
De plus elle contrôle la réponse de l'utilisateur : 0
  pour non et 1 pour oui.
Elle n'est pas en soit nécessaire comme fonction et on
  aurait pu l'inclure directement dans
la fonction releve, mais elle a l'avantage de nous faire
  apparaître la notion de fonction récurrente. */
variables globales :
entree() :entier
```

```
variables locales :
r :caractère
Début
lire(r) ;
selon
(r ='o')faireentree <-1;
(r ='n')faireentree <-0;
(r !='o'OU r !='n')faire
{afficher ("Vous avez saisi un mauvais caractère.");
afficher ("Recommencez s'il vous plait.");
entree <- entree () ;}
fin selon
Fin
procédure releve (l, a1, a2, b)
/*La fonction releve est la fonction qui permet de saisir
  les entrées de l'utilisateur.
  Celles-ci seront vérifiées. Elle fait appel à la fonction
  entree. */
variables globales :
l :chaîne de caractères/*le nom du lieu */
a1 :réel/*température maximale */
a2 :réel/*température minimale */
b :entier/*b=0 si l'utilisateur veut poursuivre; 1 autrement
  */
variables locales :
flag :entier/*drapeau permettant de stopper la boucle
  tant_que */
repoui <- 'o',repon <- 'n' :caractère
Début
afficher ("Saisie des relevés de températures. Voulez-vous
  continuer?");
afficher (repoui/"repon);
b <- entree() ;/*appel à la fonction entrée */
TANT_QUE(b =1ET flag =1)
afficher ("Donnez un relevé de température");
afficher ("Lieu :");
lire (l);
afficher ("Donnez la température maximale :");
lire (a1);
afficher ("Donnez la température minimale :");
lire (a2);
si(a1 < a2 )/*on teste si l'utilisateur à rentré les températures
  dans le bon ordre*/
```

```
alors
{afficher ("Vous donnez une température maximale inférieure
  à la température minimale.");
afficher ("Cherchiez-vous l'écran bleu?");}
sinonflag <-0;
finsi
si(a1 < zero OU a2 < zero)/*on teste sur les limites physiques
  du système*/
alorsafficher ("Changez de thermomètre : vous êtes en-dessous
  du zéro absolu!");
sinonflag <-0;
finsi
FIN_TANT_QUE
Fin
procédure calcul(Max ,tempmax, tempmin, ecart, N)
/*La procédure calcul prend en charge tous les calculs
  réalisés sur les donnés rentrées
par l'utilisateur. Elle peut faire appel à d'autres fonctions
  (par exemple ici la fonction arr()).
Nous évitons par la création d'une telle procédure de
  faire des calculs dans la fonction principale,
ainsi des modifications sur les calculs etc entraînent
  uniquement des modifications sur cette procédure
et non sur la fonction principale. */
variables globales :
Max :réel/*température maximale parmi tous les relevés
  entrés*/
tempmax :réel/*température maximale du dernier relevé
  */
tempmin :réel/*température minimale du dernier relevé
  */
ecart :réel/*différence de température entre le maximum
  Max et le minimum tempmin */
N :entier/*compteur pour connaître combien de lieu on
  atteint la température maximum Max */
variables locales :
Début
selon
( Max < tempmax )faire
{Max <- tempmax;/*on initialise Max au nouveau maximum
  */
N <-1;}/*on remet le compteur à 1 */
( Max = tempmax )faire N <- N + 1;/*on incrémente N */
( Max > tempmax )faire/*Rien à faire*/
```

```
fin selon
si( ecart < (tempmax - tempmin) )
alors ecart <- tempmax - tempmin; /*on calcul le nouvel
    écart s'il y a lieu */
fin si
Fin
procédure sortie (a,intb,intflag)
/*La fonction sortie permet d'afficher le résultat désiré
    à l'utilisateur. */
variables globales :
a :réel/*écart maximale de température */
b :entier/*nombre de lieu ayant atteint la température
    maximun */
flag :entier/*drapeau permettant de savoir si l'utilisateur
    à rentré des données
    (valeur égale à 1) ou non (valeur égale à 0) */
variables locales :
Début
si( flag =1)
alors
{afficher("Ecart de températures maximal : "a);
afficher("Arrondi de l'écart maximal de températures : "arr(a));
afficher("Nombre de lieu ayant observé la température
    maximale : "b);}
sinon afficher ("Pas de relevé.");
fin si
Fin
/*Et enfin écriture de la fonction principale main : */
fonction principale main()
/*Ce petit programme demande à l'utilisateur de rentrer
    un à un des relevés de température
    c'est à dire le nom de la ville, sa température maximun
    et sa température minimun. Ensuite il calcul
    l'écart maximun et son arrondi sur l'ensemble des villes
    entre la température maximale et la
    température minimale.
    Il compte aussi le nombre de ville ayant subit la température
    maximale. */
variables globales :
variables locales :
TMax <-zero :réel/*température maximale sur l'ensemble
    des relevés */
ecartmax <--1 :réel/*écart entre Tmax et tempmin */
```

```
tempmax :réel/*température maximale lors d'un relevé */
tempmin :réel/*température minimale lors d'un relevé */
nblieumax :entier/*nombre de lieu connaissant la température
    maximun Tmax */
drap <-1 :entier/*drapeau pour connaître a volonté de
    l'utilisateur : continuer (1) ou arrêter (0) */
k <-0 :entier/*compteur */
lieu :chaîne de caractères/*nom du lieu du relevé de température
    */
Début
afficher ("Ce programme calcul l'écart entre la température
    maximale et la température minimale pour une\
ville donnée et renvoie l'écart maximal observé sur l'ensemble
    des villes observées.\
Il renvoie aussi le nombre de lieu ayant observée la température
    maximale.");
afficher ("Les températures doivent être données en °C.");
TANT_QUE(drap =1)
k <- k + 1;
releve (lieu,tempmax,tempmin,drap);
si( drap =1)
alors calcul(TMax,tempmax,tempmin,ecartmax,nblieumax);
finsi
FIN_TANT_QUE
si(k !=1)
alors
{drap <-1;
sortie(ecartmax,nblieumax,drap);}
sinon
{drap <-0;
sortie(ecartmax,nblieumax,drap);}
finsi
Fin
```

Vous pouvez si vous le souhaitez [télécharger](http://file.trustonme.net/data/code.c.tar.gz)¹ le code en C de cet algorithme. Copiez le code dans un fichier code.c (ou décompressez le fichier code.c.tar.gz). La compilation se résume alors à :

```
gcc -Wall -c -ansi -g code.c
gcc code.o -o code.exe -lm
```

Cela va créer, un fichier code.exe qui est l'exécutable que vous pourrez exécuter par :

```
./code.exe
```

¹<http://file.trustonme.net/data/code.c.tar.gz>

Notez, que le .exe n'est absolument pas nécessaire, l'extension d'un fichier n'induit pas son caractère exécutable. Elle est juste là, par commodité et à titre pédagogique.
Suivant »»²

²<http://www.truostonme.net/didactels/150.html>