DM1 Info : consignes

À propos de ce devoir

Fichiers

Ce devoir est à déposer sous forme d'archive sur le site `Cahier de prépa` avant le vendredi 22 septembre minuit. Aucun rendu par mail n'est accepté.

Dans le menu `Informatique` de `Cahier de prépa`, cliquer sur `Transfert de document`. Une page titrée DM1 apparaît.

Il y a un seul fichier à glisser dans l'archive, il s'agit de `dm1.c`. Si une question demande autre chose que l'écriture d'un code, mettre la rédaction de sa réponse en commentaire.

Les fichiers sont inclus dans une archive `.zip` respectant la règle de nomenclature décrite ci-après :

L'élève Pierre Dupont doit constituer une archive `pierre_dupont.zip` contenant ses deux fichiers.

On autorise les réponses par binôme. Si Pierre Dupont et Marie-Chloé Léonidas de la Tour s'entendent bien, ils renvoient un fichier `marie_chloe_leonidas_de_la_tour_ET_pierre_dupont.zip`. Pas d'accent ni de trait d'union.

Pour compresser au format `.zip` le répertoire `Acompresser` et son contenu dans l'archive `pierre_dupont.zip`, se placer dans le répertoire parent et entrer :

```
zip pierre_dupont.zip Acompresser/*
```

Nom(s), email(s), compilation

En commentaire, au début de tous vos fichiers source, vous écrivez les noms des auteurs séparés par des virgules et les emails de chacun (voir figures 1 et 2).

En commentaire, au début de votre fichier source `.c`, vous écrivez la commande de compilation du fichier.

Un fichier qui ne passe pas l'épreuve de la compilation avec `-Wall -Werror=vla` est noté E. Les `Warning` font baisser la note.

```c
// Nom : Pierre Dupont
// email : pierre.dupont@gmail.com
// gcc dm1.c -Wall -Werror=vla -o exo1

```

FIGURE 1 – Pierre a fait son devoir seul

```
/*
Nom : Marie-Chloé Léonidas de la Tour; Pierre Dupont
email : pierre.dupont@gmail.com; marie.ldelatour@hotmail.fr
compilation :  gcc dm1.c -Wall -Werror=vla -o exo1
*/


```

FIGURE 2 – Pierre et Marie ont travaillé ensemble

Pas de saisie

Aucun `scanf` dans le code rendu.

Tests et `main`

Pour chaque fonction demandée en C, il faut introduire une procédure de tests unitaires correspondante. Par exemple, si on demande à Pierre Dupont d'écrire une fonction de calcul du minimum de deux entiers, Pierre rend une fonction et une procédure en essayant d'être le plus exhaustif possible dans ses tests (cf figure 3).

Observer que le code ne fait pas appel à la fonction `scanf` et que le `main` est presque vide : les tests sont effectués ailleurs.

D'une façon générale, éviter les `scanf` pour mes devoirs.

Commentaires

Un code lisible est un code commenté : la raison de telle instruction conditionnelle ou telle boucle est expliquée. On évite de paraphraser le code.

Les variables locales ont des noms courts qui expliquent leur signification (`n` pour un entier, `cpt` pour un compteur...).

Un code long doit être factorisé en sous-fonctions aux noms explicites.

Note

Les devoirs maisons sont toujours corrigés et une note indicative (A,B,C,D,E) est fournie. Ces notes ne sont pas prises en compte dans le bulletin. Il ne faut donc pas s'alarmer si votre performance n'est pas bonne.

Il est totalement improductif de recopier la réponse d'un camarade plus à l'aise sauf si cherchez à m'agacer : vous n'apprendrez rien de la sorte et vous me ferez perdre mon temps.

Si vous n'arrivez pas à tout faire, ne faites pas tout. Rendez un travail personnel (on apprend en réfléchissant, pas en recopiant) de façon à ce que vos points forts soient soulignés et les points à travailler, identifiés.

Songez qu'il existe des outils de comparaison de fichier. Si une fraude est avérée, elle peut être mentionnée dans le bulletin.

```c
// Nom : Pierre Dupont
// email : pierre.dupont@gmail.com
// gcc mini.c -Wall -o minimum

# include <stdio.h>

// Q1.a
int mini(int x, int y){// fonction mini : but de l'exo 1
  if (x < y)
    return x;
  else return y;
}

void test_mini(){// fonction de tests de mini
  int x = 2, y =3;// x<y
  printf("mini(%d,%d)=%d\n", x,y,mini(x,y));

  x=3;y=2;// x>y
  printf("mini(%d,%d)=%d\n", x,y,mini(x,y));

  x=3,y=3;// x=y
  printf("mini(%d,%d)=%d\n", x,y,mini(x,y));

  //Aucun scanf svp !!!
}

/* Q1.b
La complexité est en O(1) car il n'y a
ni boucle, ni appel récursif ni appel de fonction.
*/

int main(void){
  test_mini();// le main est presque vide; c'est voulu !
  //test_addition_matrice(); (test désactivé)
  return 0;
}
```

FIGURE 3 – Le code de Pierre Dupont pour un exercice en C