


## Algorithmique et programmation (I3) contrôle

Durée : 1 heure 20 minutes. Aucun document autorisé.

### 1 Étude et modification de programme (5 points)

Le programme suivant comporte deux erreurs :

 15 min

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int factorielle(int n);
5
6  int main () {
7      /* test de factorielle */
8      printf("Test (doit donner 120) : %d\n", factorielle(5));
9
10     return EXIT_SUCCESS;
11 }
12
13 double factorielle(int n) {
14     int res;
15     int i;
16
17     i = 1;
18     while (i <= n) {
19         res = res * i;
20         i = i + 1;
21     }
22
23     return res;
24 }
```


**Question A.** L'une des erreurs empêche la compilation du programme. Corriger.

**Question B.** L'autre erreur génère un calcul faux. Corriger.

**Question C.** Lignes 17 à 21, il aurait été plus naturel d'employer une autre structure de contrôle que le `while`. Quelle structure, et qu'écririez-vous à la place de ces lignes ?

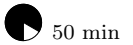
### 2 Sans fonctions (5 points)

**Question D.** Soit un tableau `t` d'entiers, que vous initialiserez à des valeurs de votre choix, d'une taille donnée par une constante symbolique `N`. Écrire un programme qui demande un entier `seuil` à l'utilisateur puis affiche les éléments de `t` supérieurs ou égaux à ce seuil. Par exemple, si les éléments de `t` sont les entiers 4, 6, -1, 6, 7, 4 et si l'utilisateur fixe le `seuil` à 6 alors le programme affichera 6, 6, 7.

 15 min

### 3 Canard aux olives (10 points)

Vous disposez d'un recueil de recettes avec, pour chaque recette, un temps de préparation et un temps de cuisson. Pour les viandes le temps de cuisson est donné par kilogramme. Vous souhaitez produire une application qui calculera la durée réelle de cuisson et la durée totale de réalisation de la recette (préparation et cuisson). Aujourd'hui, il vous faut mettre au point la partie affichage et calcul des durées, en travaillant sur un exemple de recette.



**Question E.** Déclarer un type `duree_t` permettant de stocker des durées exprimées en heures, minutes, secondes (valeurs entières) et définir une procédure `afficher` permettant d'afficher une durée, comme dans l'exemple.

**Question F.** Déclarer et définir une fonction `normaliser` mettant une durée sous la forme usuelle où le nombre de secondes et le nombre de minutes sont inférieurs à 60. Par exemple, la durée 1h 75min 64sec sera normalisée en 2h 16min 4sec.

**Question G.** Déclarer et définir une fonction `add` permettant d'additionner des durées et une fonction `mult` permettant de multiplier une durée par un facteur fourni en paramètre (un double). Ces fonctions rendront des résultats normalisés.

**Question H.** Compléter la partie calcul du programme.

```
int main() {
    duree_t prep = {0,30,0}; /* tps de preparation 30 minutes */
    duree_t par_kg = {0,52,0}; /* tps de cuisson par kilo 52 minutes */
    double poids = 1.94; /* poids du canard 1.94 kg */

    duree_t cuisson; /* temps reel de cuisson, à calculer */
    duree_t realisation; /* temps total de realisation, à calculer */

    printf("***Recette du canard aux olives.**");
    printf("\nTemps de préparation : ");
    afficher(prepare);
    printf("\n\n* Temps de cuisson ");
    printf("\nPar kg : ");
    afficher(par_kg);

    /* ici il faut ajouter le calcul des durees reelles */

    printf("\nPour un canard de %lg kg : ", poids);
    afficher(cuisson);
    printf("\nTemps total de la recette : ");
    afficher(realisation);

    return EXIT_SUCCESS;
}
```

Exemple d'affichage.

```
**Recette du canard aux olives.**
Temps de préparation : 0h 30min 0sec

* Temps de cuisson
Par kg : 0h 52min 0sec
Pour un canard de 1.94 kg : 1h 40min 52sec
Temps total de la recette : 2h 10min 52sec
```