
Éléments d'informatique, rattrapage

Durée : 3 heures.

Documents autorisés : Aucun.

Recommandations : Un barème vous est donné à titre indicatif afin de vous permettre de gérer votre temps. La notation prendra en compte à la fois la syntaxe et la sémantique de vos programmes, c'est-à-dire qu'ils doivent compiler correctement. Une fois votre programme écrit, il est fortement recommandé de le faire tourner à la main sur un exemple pour s'assurer de sa correction.

1 Somme des éléments d'un tableau (*4 points*)

Nous voulons écrire un programme qui calcule la somme des éléments d'un tableau d'entiers. Une partie du programme est déjà écrite, et Pippo pense qu'il ne reste plus qu'à écrire la partie qui calcule effectivement la somme. Voici son programme :

```
1  #include <stdlib.h> /* EXIT_SUCCESS */
2  #include <stdio.h> /* printf, scanf */
3
4  /* déclaration constantes et types utilisateurs */
5  #define TAILLE 3
6
7  /* Fonction principale */
8  int main()
9  {
10     int t[TAILLE] = {12,10,15}; /* tableau à sommer */
11
12
13
14     /* calcul de la somme (À FAIRE) */
15
16
17
18
19
20     /* affichage du résultat */
21     printf("La somme est %d\n", somme);
22
```

```

23     /* valeur fonction */
24     return EXIT_SUCCESS;
25 }

```

1. La compilation échoue. Expliquer pourquoi, quelle étape de la compilation échoue précisément et ce qu'il manque pour que la compilation réussisse.
2. Expliquer le fonctionnement de l'instruction `#define`.
3. Compléter le programme pour qu'il calcule effectivement la somme des éléments du tableau (pour une taille de tableau arbitraire).

Un exemple de sortie du programme pour le tableau donné dans le code est :

La somme est 37

2 Sans fonctions

Il est demandé de résoudre les deux problèmes suivants sans définir de fonctions utilisateurs. L'ensemble du code sera à écrire dans la fonction principale `main`.

2.1 For ou while (4 points)

Écrire le programme qui :

- demande à l'utilisateur d'entrer les valeurs d'un tableau d'entiers de taille N (une constante symbolique)
- puis cherche si le tableau contient l'entier zéro et affiche l'indice de la première case nulle s'il y en a une ou **Pas de case nulle** sinon.

Deux exemples d'exécution sont les suivants (pour N valant 4) :

```

Saisissez 4 entiers : 8 31 1 -12
Pas de case nulle

```

```

Saisissez 4 entiers : 2 -5 4 0
Case nulle d'indice 3

```

2.2 Bientôt l'été (5 points)

Écrire un programme qui demande à l'utilisateur de saisir une date dans l'année et affiche :

- **C'est le printemps**, si la date est au printemps ;
- **Déjà l'été** si la date du jour est en été ;
- **Ce n'est ni le printemps, ni l'été**, si la date n'est ni au printemps ni en été ;

Vous utiliserez trois constantes symboliques pour représenter les trois périodes différentes de l'année : `PRINTEMPS`, `ETE`, `AUTRE`.

Vous séparerez clairement la partie saisie, la partie calcul de la saison et la partie affichage.

La date sera saisie sous la forme de deux entiers : l'un entre 1 et 31 (inclus) pour le jour dans le mois et l'autre entre 1 et 12 (inclus) pour le numéro du mois. On suppose que la saisie est une date correcte.

Le printemps commence le 20 mars, l'été le 21 juin, l'automne le 22 septembre.

Exemples d'exécutions :

Date : 21 10
Ce n'est ni le printemps, ni l'ete

Date : 11 6
C'est le printemps

Date : 21 6
C'est l'ete

3 Trace d'un programme avec fonctions (*3 points*)

Simulez l'exécution du programme suivant, en réalisant sa **trace**, comme cela a été vu en TD et en cours.

```
1  #include <stdlib.h> /* EXIT_SUCCESS */
2  #include <stdio.h> /* printf, scanf */
3
4  /* declarations constantes et types utilisateurs */
5
6  /* declarations de fonctions utilisateurs */
7  int mystere(int n);
8
9  int main()
10 {
11     int x = 2;
12     int res;
13
14     res = mystere(x);
15     printf("mystere(%d) = %d\n", x, res);
16
17     return EXIT_SUCCESS;
18 }
19
20 /* definitions de fonctions utilisateurs */
21 int mystere(int n)
22 {
23     if (n > 1)
24     {
25         return n * mystere(n - 1);
26     }
27     return 1;
28 }
```

4 Écriture de fonctions (*4 × 1 point*)

1. Écrire une fonction qui prend en entrée un entier n et calcule la somme :

$$\sum_{k=1}^n \frac{1}{k^2}$$

2. Écrire la procédure `rect` qui n'a pas d'entrée et affiche le motif suivant :

```
*****  
*      *  
*****
```

3. Écrire la procédure `rectangle` qui prend en entrée un entier `largeur` et un entier `hauteur` et affiche un rectangle plein d'étoiles de `hauteur` lignes sur `largeur` colonnes.
4. Écrire une fonction `volume_cylindre` qui prend en entrée un rayon réel `r` et une hauteur réelle `h` et renvoie le volume du cylindre de rayon `r` et de hauteur `h`. vous ferez appel à une fonction `calcul_surface`, supposée déjà définie, qui calcule la surface d'un cercle à partir de son rayon.

Bonus. Écrire une procédure `boite` qui affiche un rectangle de `hauteur` lignes sur `largeur` colonnes, comme la fonction `rectangle` (question 3), mais creux, comme dans l'exemple de la question 2.