

Examen, 2h, avec documents

NOM :

Prénom :

Consignes : Téléphones, ordinateurs et calculatrices interdits.

Documents de cours/TD/TP autorisés.

Aucune question possible : en cas d'ambiguïté dans le sujet, signaler le choix fait sur la copie.

Conseil : Lire complètement le sujet avant de commencer à y répondre.

Barème indicatif :

exercice	1	2	3
points	5	5	10

Exercice 1 Répondre de manière concise et précise aux questions suivantes.

1. On considère le sous-programme suivant :

```
1 int * sp() {
2     int tab[10];
3     int i = 0;
4     while (i <= 10) {
5         tab[i] = i * i;
6         i++;
7     }
8     return tab;
9 }
```

Il contient au moins deux maladroresses et deux erreurs. Lesquelles ? Justifier les réponses et corriger le programme.

2. Expliquer la différence entre taille effective et capacité d'un tableau.

3. Le terme sous-programme est le terme général qui désigne les procédures et les fonctions.

3.1. Indiquer l'intérêt d'un sous-programme.

3.2. Expliquer la différence entre procédure et fonction.

3.3. À quoi reconnaît-on une procédure en C ?

4. Expliquer les notions de *paramètre formel* et de *paramètre effectif*.

Exercice 2 : Spécifier des sous-programmes

Pour chacun des énoncés suivants, donner la spécification du sous-programme correspondant.

1. Calculer la puissance entière d'un réel
2. Saisir un entier compris entre une borne inférieure et une borne supérieure. Avant chaque demande à l'utilisateur, une consigne lui est affichée.
3. Savoir si une année est bissextile
4. Calculer le pgcd de deux entiers strictement positifs
5. Obtenir le quotient et le reste d'une division entière

Exercice 3 : Modéliser un robot de type 1

L'objectif de cet exercice est de modéliser un robot de type 1. Un tel robot se déplace dans un environnement qui peut être modélisé par un quadrillage. Chaque case du quadrillage correspond à une position possible du robot. Elle est repérée par son abscisse (x) et son ordonnée (y).

Le robot de type 1 est caractérisé par sa position (abscisse et ordonnée) dans l'environnement et sa direction. Seulement quatre directions sont possibles : haut, bas, droite ou gauche.

Un robot de type 1 peut avancer dans son environnement d'un pas (d'une case) suivant sa direction et il peut pivoter de 90° sur sa droite.

1. Définir le type Robot1 qui définit un robot de type1. Il permettra par exemple de déclarer une variable représentant un robot positionné à l'abscisse 2 et ordonnée 5 et dans la direction « gauche ».
2. Écrire un sous-programme qui crée un robot de type 1 à partir de sa position initiale (abscisse et ordonnée) et de sa direction initiale. Par exemple, on peut créer un robot d'abscisse 2, ordonnée 5 et direction « gauche ».
3. Écrire un sous-programme qui fait pivoter un robot de 90° vers la droite.
4. Écrire un sous-programme qui fait avancer un robot d'une seule case suivant sa direction courante.
5. Écrire un programme correspondant au scénario suivant.
 1. Créer un robot r1 en position (4,10) orienté « droite ».
 2. Créer un robot r2 en position (15,7) orienté « bas ».
 3. Faire pivoter le robot r1.
 4. Vérifier que la direction de r1 est « bas ».
 5. Faire avancer le robot r2.
 6. Vérifier que l'ordonnée de r2 est 6.
6. Expliquer comment faire un module robot1 à partir des éléments écrits ci-dessus.