UNIVERSITÉ DE PERPIGNAN

FILIÈRE :

ÉPREUVE :

du :

Nom et Prénoms du Candidat :

N°

Ne pas dégrafer les feuilles.

Durée : 2 heures. Aucun document autorisé.

Modalités : Répondre uniquement dans les cadres prévus à cet effet. La qualité de la rédaction sera prise en compte dans la notation. Un barème est indiqué à droite de chaque question. La note finale sur 20 est proportionnellement déduite de ce barème.

Exercice 1. (22 points) **/22**

1. Écrire le corps de la procédure suivante qui permute (échange) les valeurs de `a` et `b` :

   `fonction swap(a : in out entier, b : in out entier)` **/1**

2. Écrire un algorithme qui utilise `swap` pour permuter puis afficher deux valeurs entrées au clavier. **/1**

On souhaite inverser l'ordre des valeurs d'un tableau donné (taille et valeurs connues) : le tableau $[a_0, a_1, ..., a_{n-1}]$ sera renversé en $[a_{n-1}, ..., a_1, a_0]$.

3. Donner le principe d'un algorithme qui réalise cette inversion en utilisant la procédure `swap`. Bien différencier les cas selon la parité de la longueur n du tableau pour identifier le parcours à effectuer. **/2**

4. Écrire un algorithme qui réalise cette inversion. Le tableau sera par exemple de taille n=10 et initialisé à votre goût. **/3**

5. Écrire l'en-tête d'un sous-programme (fonction ou procédure) `renverse` qui effectue le traitement précédent sur un tableau passé en paramètre. **/1**

6. Écrire un algorithme qui utilise `renverse` pour inverser l'ordre des valeurs d'un tableau entrées au clavier. Le tableau inversé sera affiché. **/2**

7. Écrire le corps du sous-programme `renverse`. **/1**

8. Proposer un paramètre de complexité pour `renverse` ainsi que l'expression de sa fonction de complexité. **/2**

9. Caractériser la complexité asymptotique de `renverse`. Donner un exemple d'algorithme de complexité asymptotique similaire. **/2**

10. En supposant que `swap` est correcte, justifier qu'un invariant de boucle permet de prouver la correction de `renverse`. **/1**

11. Proposer un invariant de boucle qui permet de prouver la correction de `renverse`. **/3**

12. Prouver cet invariant. **/3**

Exercice 2. (10 points) **/10**

Les coefficients binomiaux $\binom{n}{p}$, $0 \leq p \leq n$, utiles au calcul de $(a+b)^n$ sont définis comme suit :

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}.$$

Ils sont aussi calculables avec la relation de Pascal :

$$\binom{n}{0} = \binom{n}{n} = 1 \text{ et } \binom{n}{p} = \binom{n-1}{p-1} + \binom{n-1}{p} \text{ pour } 0 < p < n.$$

1. Ecrire la fonction récursive `f` qui calcule la fonction factorielle. Bien préciser en-tête et corps. **/2**

2. Écrire (l'en-tête et le corps) d'une fonction `binom` qui calcule $\binom{n}{p}$ pour deux valeurs entières n et p. La valeur non significative -1 sera retournée le cas échéant. **/2**

3. Écrire un algorithme qui calcule et affiche $\binom{n}{p}$ pour deux valeurs n et p entrées au clavier. **/1**

4. Expliciter la suite d'appels nécessaires au calcul de $\binom{4}{3}$. Comptabiliser le nombre total d'appels à `f`. Comptabiliser et expliciter les appels éventuellement répétés. Qu'en penser ? **/3**

5. Expliciter et commenter deux principes du calcul de $\binom{n}{k}$ grâce à la relation de Pascal. **/2**