

NOM et Prénom :

Jeudi 26 septembre 2024

Term Spé N.S.I.

Interrogation (55 min.)

(Calculatrice non autorisée)

Exercice 1 – Tri par sélection (6,5 points)

1° Compléter la fonction `tri_selection(tab)` ci-dessous :

```
def tri_selection(tab) :  
  
    """  
  
    Tri d'une liste par l'algorithme : Tri par sélection  
  
    ATTENTION : La liste en paramètre est modifiée  
  
    """  
  
    n = len(tab)  
  
    for i in range(n-1) :  
  
        indice_mini = . . .  
  
        valeur_mini = . . .  
  
        valeur_debut = . . .  
  
        for j in range(. . ., n) :  
  
            valeur = tab[j]  
  
            if valeur < valeur_mini :  
  
                valeur_mini = . . .  
  
                indice_mini = . . .  
  
        if indice_mini != i :  
  
            tab[indice_mini] = valeur_debut  
  
            tab[i] = valeur_mini
```

2°) On appelle `tri_selection([3, 9, 2, 4, 1, 7])`, donner l'état des différentes variables et du test (True ou False) de la boucle principale lors de l'exécution de cette fonction :

AVANT exécution de la boucle sur j			APRES exécution de la boucle sur j		
i	tab	valeur_debut	indice_mini	valeur_mini	Test : indice_mini != i
0	[3, 9, 2, 4, 1, 7]	3	4	1	True
1	[..., ..., ..., ..., ..., ...]				
2	[..., ..., ..., ..., ..., ...]				
	[..., ..., ..., ..., ..., ...]				
	[..., ..., ..., ..., ..., ...]				
FIN :	[..., ..., ..., ..., ..., ...]				

Exercice 2 – Tri par insertion (6,5 points)

1°) Compléter la fonction `tri_insertion(tab)` ci-dessous :

```
def tri_insertion(tab) :  
    """  
    Tri d'une liste par l'algorithme : Tri par insertion  
    ATTENTION : La liste en paramètre est modifiée  
    """  
    n = len(tab)  
    for i in range(1, n) :  
        valeur = tab[i]  
        j = . . .  
        while . . . and . . . :  
            tab[j+1] = . . .  
            tab[j] = . . .  
            j -= 1
```

2°) On appelle `tri_insertion([3, 9, 2, 4, 1, 7])`, donner l'état des différentes variables et de la condition d'arrêt de la boucle `while` (True ou False) lors de l'exécution de cette fonction :

BOUCLE FOR sur i			BOUCLE WHILE sur j		
i	tab	valeur	j	tab[j]	Condition d'arrêt de la boucle WHILE
1	[3, 9, 2, 4, 1, 7]	9	0	3	False
2	[..., ..., ..., ..., ..., ...]				
3	[..., ..., ..., ..., ..., ...]				
4	[..., ..., ..., ..., ..., ...]				
5	[..., ..., ..., ..., ..., ...]				
FIN :	[..., ..., ..., ..., ..., ...]				

Exercice 3 – Recherche dichotomique (7 points)

1°)

```
def dichotomie(tab, valeur) :  
  
    """  
  
    Recherche dichotomique de la valeur dans tab  
  
    ATTENTION : tab doit être trié dans l'ordre croissant  
  
    """  
  
    indice_min = 0  
  
    indice_max = len(tab) - 1  
  
    while . . . :  
  
        indice_milieu = . . .  
  
        valeur_milieu = . . .  
  
        if valeur == valeur_milieu :  
  
            return . . .  
  
        elif valeur < valeur_milieu :  
  
            indice_max = . . .  
  
        else :  
  
            indice_min = . . .  
  
    return . . .
```

2°) On appelle **dichotomie**([1, 3, 4, 6, 8, 9, 10, 12, 13, 16, 18], 11), donner l'état des différentes variables, de la condition d'arrêt de la boucle **while** (True ou False) ainsi que le test qui est validé dans le **if ... elif ... else** : ... lors de l'exécution de cette fonction :

indice_min	indice_max	Condition d'arrêt de la boucle WHILE	indice_milieu	valeur_milieu	Test validé (if, elif ou else)
0	10	True	5	9	else
Valeur renvoyée par la fonction :					

3°) On appelle **dichotomie**([1, 3, 4, 6, 8, 9, 10, 12, 13, 16, 18], 3), donner l'état des différentes variables, de la condition d'arrêt de la boucle **while** (True ou False) ainsi que le test qui est validé dans le **if ... elif ... else** : ... lors de l'exécution de cette fonction :

indice_min	indice_max	Condition d'arrêt de la boucle WHILE	indice_milieu	valeur_milieu	Test validé (if, elif ou else)
0	10	True	5	9	elif
Valeur renvoyée par la fonction :					