

Aucun document ou support autre que le sujet ou les copies d'examen n'est autorisé.  
 (la copie ou les brouillons du voisin ne sont pas des supports autorisés).  
 Éteignez impérativement vos mobiles.

Lorsque des calculs sont nécessaires, il est impératif de les présenter sur la feuille d'examen. Il est aussi nécessaire de justifier ses réponses.

## 1 Exercice

Un palindrome est un mot qui contient la même suite de lettres qu'il soit parcouru de gauche à droite ou de droite à gauche. *Kayak* ou *ressasser* sont des palindromes. Les mots d'une seule lettre sont évidemment des palindromes et on admettra que le mot vide est un palindrome.

1. Complétez le programme suivant de sorte qu'il affiche `palindrome` si le contenu de la pile `p1` constitue un palindrome et `pas palindrome` sinon.

```
import java.util.*;

public class Pal {
    public static void main(String []a) {
        Stack<Character> p1 = new Stack<Character>();
        Stack<Character> p2 = new Stack<Character>();
        Stack<Character> p3 = new Stack<Character>();
        p1.push('k'); p1.push('a'); p1.push('y'); p1.push('a'); p1.push('k');
        // ou p1.push('a'); p1.push('a'); p1.push('k');
        boolean b = true;

        // à compléter...

        if (b) System.out.println("palindrome");
        else System.out.println("pas palindrome");
    }
}
```

2. Si la pile `p1` contient au départ  $n$  lettres, combien d'opérations `push`, `pop` et `isEmpty` sont exécutées ?

## 2 Exercice

Soit le programme suivant :

```
public class Rec3 {
    public static int f(int n) {
        if (n<=1) return 1;
        return n*f(n-2)*(n-1);
    }
    public static void main(String []a) {
        for (int i=0; i<6; i++) {
            System.out.println(f(i));
        }
    }
}
```

1. Qu'affiche ce programme lors de son exécution ?
2. Quelle fonction mathématique calcule la fonction Java `f` ? Justifiez.
3. Pour une valeur  $n \geq 0$ , combien d'appels récursifs sont-ils réalisés ? Justifiez.
4. Combien de variables ce programme utilise-t-il ? Précisez la nature et le type de ces variables.
5. Pour un appel à `p(10)` combien de mémoire sera utilisée ? (Comptez le nombre d'entiers nécessaires).
6. Traduisez ce programme tel que vu en cours, c'est-à-dire en utilisant un tableau d'entiers statique, une variable contenant le numéro d'instruction courante, un seul `switch`, un seul `while`, etc.

### 3 Exercice

Soit le programme suivant :

```
public class Rec2 {
    public static int p(int n,int m) {
        if (n==m) return n;
        if (n<m) return p(n,m-n);
        else return p(n-m,m);
    }
    public static void main(String []a) {
        System.out.println(p(6,15));
    }
}
```

1. Qu'affiche le programme à l'exécution ?
2. Ce programme est récursif mais est-il récursif terminal? Pourquoi?
3. Dérécursivez ce programme comme fait en cours.

### 4 Exercice

Soit le programme java suivant :

```
public class Rec {
    public static int b(int n,int k) {
        if (k<0) return 0;
        if (n==0) return 1;
        return b(n-1,k+1)+b(n-1,k)+b(n-1,k-1);
    }
    public static void main(String []a) {
        System.out.println(b(3,2));
    }
}
```

1. Qu'affiche le programme précédent à l'exécution ?
2. Indiquez combien d'appels récursifs sont réalisés par le calcul ? Dessinez l'arbre des appels correspondant.
3. Si  $k > n$  au premier appel, combien d'appels récursifs sont-ils effectués (en fonction de  $k$  et  $n$ ) ?
4. Écrivez le code correspondant au calcul de la même fonction en utilisant la méthode de mémorisation.
5. Si  $k > n$  au premier appel, combien d'appels récursifs sont-ils effectués (en fonction de  $k$  et  $n$ ) ?