

NOM :	Prénom :	N° :
Examen	Concepts Informatiques (CI2)	16/05/2017
Aucun document – Aucune machine		09 <sup>00</sup> – 11 <sup>00</sup>

1. Considérons l'expression

$(5 / (3 - 1)) * (2 + 4).$

Dessiner son arbre syntaxique.

Donner sa forme préfixe.

Donner sa forme postfixe.



Décrire l'évolution du contenu de la pile lors de l'évaluation de sa forme postfixe.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Décrire l'évolution du contenu de la pile lors de la transformation de sa forme infixe en sa forme postfixe.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Décrire l'évolution du contenu de la pile lors de l'évaluation de sa forme préfixe.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

2. On considère le programme ci-dessous.

Décrire l'affichage produit par son exécution.

```
class H{
    public static int s = 3;

    public static int g(int a, int b){
        System.out.println("g("+a+", "+b+"");
        do a += b*s;
        while(a < b);
        return a-b;
    }
    public static int h(int c){
        System.out.println("h("+c+"");
        if(c < 13) return g(c/s,c)/2;
        return h(c/2);
    }
    public static void main(String[] args){
        System.out.println(h(13));
        System.out.println(h(33));
    }
}
```

Le traduire (comme en cours).

```
import java.util.*;  
class HTraduit{
```

3. On considère le programme ci-dessous.

```
1  class Wibo{  
2      public static void main(String[] args){  
3          int k=9;  
4          System.out.println("wibo("+k+")="+wibo(k));  
5          // System.out.println("wiboDyna("+k+")="+wiboDyna(k));  
6          // System.out.println("wiboTerm("+k+")="+wiboTerm(k));  
7          // System.out.println("wiboIter("+k+")="+wiboIter(k));  
8      }  
9      static int wibo(int n){  
10         if (n<=0) return 1;  
11         if (n==1) return 4;  
12         if (n==2) return 3;  
13         return wibo(n-2) + wibo(n-3);  
14     }  
15 }
```

Dessiner l'arbre des appels récursifs lors de l'évaluation de la ligne 4.

Utiliser l'approche dynamique de sorte que l'on puisse décommenter la ligne 5.

Écrire une version récursive terminale (que l'on puisse décommenter la ligne 6).

En déduire une version itérative (que l'on puisse décommenter la ligne 7).