

# Corrigés des exercices sur la mémoire

## Exercice 6.1 déroulement d'une boucle for

---

```
1 class Exo4_1{
2     public static void main(String[] args){
3         int x;
4         Terminal.ecrireString("Un entier svp:");
5         x = Terminal.lireInt();
6         for (int i = 0; i<4; i++){
7             Terminal.ecrireIntln(x+i);
8         }
9         Terminal.ecrireStringln("Fini");
10    }
11 }
```

---

Voici le tableau qui retrace l'exécution de ce programme :

nb	test	x	i	clavier	écran
3		?	NEP		
4		?	NEP		Un entier svp :
5		5	NEP	5	
6.init		5	0		
6.test	true	5	0		
7		5	0		5
6.incr		5	1		
6.test	true	5	1		
7		5	1		6
6.incr		5	2		
6.test	true	5	2		
7		5	2		7
6.incr		5	3		
6.test	true	5	3		
7		5	3		8
6.incr		5	4		
6.test	false	5	4		
9		5	NEP		Fini

---

## Exercice 6.2 déroulement d'une boucle while

---

```

1  class Exo6_2{
2      public static void main(String[] args){
3          int puis = 1;
4          int x, res;
5          Terminal.ecrireString("Un entier svp:");
6          x = Terminal.lireInt();
7          res = x;
8          while (res < 1000){
9              res = res *x;
10             puis = puis+1;
11         }
12         Terminal.ecrireString("Le résultat est");
13         Terminal.ecrireInt(puis);
14         Terminal.sautDeLigne();
15     }
16 }

```

---

1. Que calcule ce programme ?

**Réponse : pour un entier x entré au clavier, ce programme calcule la plus petite puissance de x supérieure à 1000.**

2. Avec un tableau, retracez une exécution du programme suivant dans laquelle on entre au clavier la valeur 8.

lig	puis	x	res	res < 1000	écran	clavier
	NEP	NEP	NEP			
3	1	NEP	NEP			
4	1	?	?			
5	1	?	?		un entier svp :	
6	1	8	?	8		
7	1	8	8			
8	1	8	8	true		
9	1	8	64			
10	2	8	64			
8	2	8	64	true		
9	2	8	512			
10	3	8	512			
8	3	8	512	true		
9	3	8	4096			
10	4	8	4096			
8	4	8	4096	false		
12	4	8	4096		le résultat est	
13	4	8	4096		4	
14	4	8	4096			

---

## Exercice 6.3 état de la mémoire

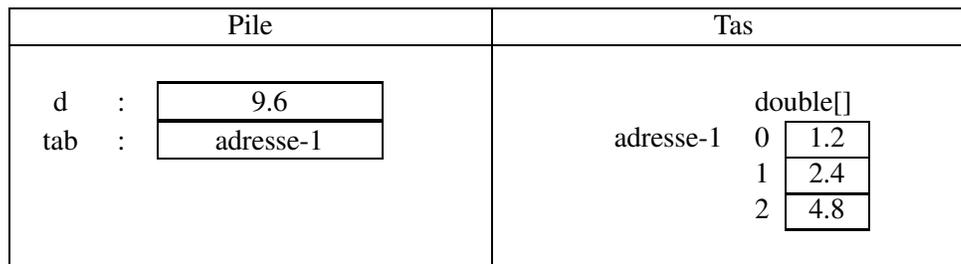
### Question 1 *programme simple*

Dessinez l'état de la mémoire à la fin de l'exécution du programme suivant (après exécution de la dernière instruction du programme).

---

```
public class TabSimple{
    public static void main(String[] args){
        double d = 1.2;
        double[] tab = new double[3];
        for (int i=0; i<3; i=i+1){
            tab[i] = d;
            d = d *2;
        }
    }
}
```

---



### Question 2 *programme un peu bizarre*

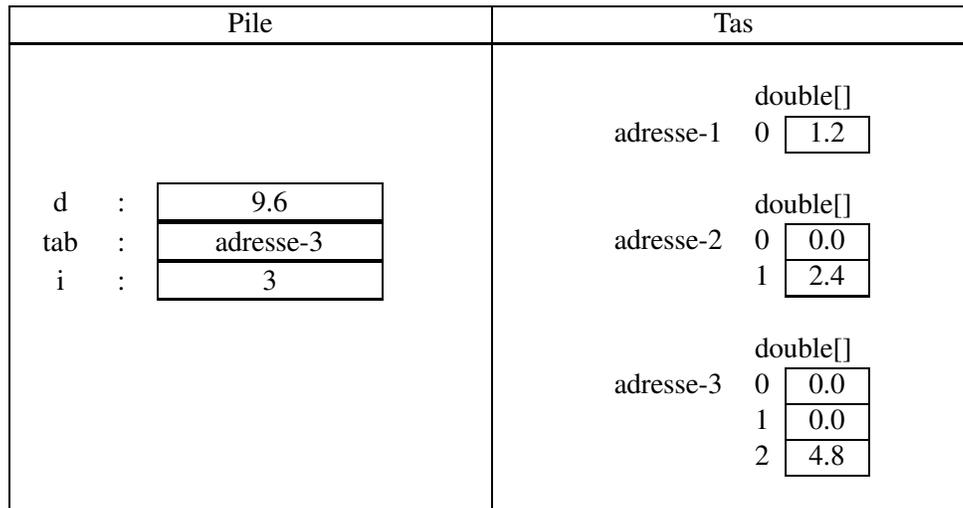
Même question pour le programme suivant. Vous supposerez ans votre dessin que les tableaux créés au cours du programme sont conservés jusqu'à la in, même s'ils ne sont plus utiles.

---

```
public class TabBiz{
    public static void main(String[] args){
        double d = 1.2;
        double[] tab;
        for (int i=0; i<3; i=i+1){
            tab = new double[i+1];
            tab[i] = d;
            d = d *2;
        }
    }
}
```

---

Notez bien qu'à chaque tour de boucle, le programme crée un nouveau tableau et n'affecte que la dernière case du tableau créé.



On n'était pas obligé de faire apparaître la variable *i* sur le dessin en considérant que la boucle terminée, celle-ci n'existe plus.

## Exercice 6.4 propriétés d'un tableau

### Question 1 *le tableau ne contient que des lettres*

```

public class QueDesLettres{
    public static void main(String[] args){
        int nbchar;
        char[] tab;
        boolean tousLettres = true;
        System.out.print("Combien_de_caractères?_");
        nbchar = Terminal.lireInt();
        tab = new char[nbchar];
        for (int i=0; i<nbchar; i=i+1){
            System.out.print("Entrez_le_caractère_numéro_" + i + ":_");
            tab[i]=Terminal.lireChar();
        }
        for (int i=0; i<nbchar; i=i+1){
            if (! Character.isLetter(tab[i])){
                tousLettres = false;
            }
        }
        if (tousLettres){
            System.out.println("Tous_les_caractères_sont_des_lettres");
        }else{
            System.out.println("Tous_les_caractères_ne_sont_pas_des_lettres");
        }
    }
}

```

---

Il suffit de trouver un seul caractère qui n'est pas une lettre pour que la propriété soit fausse. Quand on n'a encore regardé aucune case, on n'a pas encore trouvé de case sans lettre. On initialise donc la variable booléenne `tousLettres` à vrai. Puis on regarde une à une toutes les cases du tableau. Cette variable contient la valeur vraie tant qu'on n'a pas trouvé de case avec un caractère qui n'est pas une lettre.

A noter qu'on pourrait arrêter la boucle dès qu'on a trouvé un caractère qui n'est pas une lettre (on sait déjà que la propriété est fausse quel que soit le contenu des cases suivantes).

Voici la version qui arrête le parcours.

---

```
public class QueDesLettres2{
    public static void main(String[] args){
        int nbchar;
        char[] tab;
        boolean tousLettres = true;
        int idx;
        System.out.print("Combien_de_caractères?_");
        nbchar = Terminal.lireInt();
        tab = new char[nbchar];
        for (int i=0; i<nbchar; i=i+1){
            System.out.print("Entrez_le_caractère_numéro_" + i + ":_");
            tab[i]=Terminal.lireChar();
        }
        idx = 0;
        while (idx<nbchar && tousLettres){
            if (! Character.isLetter(tab[idx])){
                tousLettres = false;
            }else{
                idx = idx + 1;
            }
        }
        if (tousLettres){
            System.out.println("Tous_les_caractères_sont_des_lettres");
        }else{
            System.out.println("Tous_les_caractères_ne_sont_pas_des_lettres");
        }
    }
}
```

---

### Question 2 le tableau contient au moins une lettre

Pour un tableau de caractère entré au clavier comme à la question précédente, écrivez un programme qui détermine si le tableau contient au moins une lettre. La réponse affichée doit être oui ou non selon le cas.

---

```
public class AuMoinsUneLettre{
    public static void main(String[] args){
        int nbchar;
```

---

```

char[] tab;
boolean uneLettre = false;
System.out.print("Combien_de_caractères?_");
nbchar = Terminal.lireInt();
tab = new char[nbchar];
for (int i=0; i<nbchar; i=i+1){
    System.out.print("Entrez_le_caractère_numéro_" + i + ":_");
    tab[i]=Terminal.lireChar();
}
for (int i=0; i<nbchar; i=i+1){
    if (Character.isLetter(tab[i])){
        uneLettre = true;
    }
}
if (uneLettre){
    System.out.println("oui,_il_y_a_au_moins_une_lettre");
}else{
    System.out.println("non,_il_n'y_a_aucune_lettre");
}
}

```

---

La variable `uneLettre` contient `false` jusqu'à la première lettre rencontrée dans le tableau. Ici encore, on pourrait arrêter le parcours des cases à la première lettre rencontrée.