

**EXERCICE 3C.1**

Dans chaque cas, indiquer les valeurs successives des variables pendant l'exécution de l'algorithme, et entourer les valeurs qui s'affichent à l'écran :

1.

*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U

2.

*U prend la valeur 0*  
*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Afficher *U*  
 Fin de boucle

P	N	U

3.

*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U

4.

*U prend la valeur 0*  
*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U

5.

*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Afficher *U*  
 Fin de boucle

P	N	U

6.

*U prend la valeur 1*  
*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U

b. Parmi tous ces algorithmes, indiquer celui qui permet de calculer le plus rapidement possible  $u_3$ , où  $u_n$  est la suite

définie par récurrence par : 
$$\begin{cases} u_0 = 0 \\ u_{n+1} = 3u_n + 2 \end{cases}$$

**EXERCICE 3C.2** On considère les programmes (TI-82) suivants

**PROGRAM:SUITES1**

```
: 1 → P
: 2 → U
: Prompt N
: While P ≤ N
: 2U + 5 → U
: P + 1 → P
: Disp U
: End
```

**PROGRAM:SUITES2**

```
: 1 → P
: 2 → U
: Prompt N
: While P ≤ N
: 2U + 5 → U
: P + 1 → P
: End
: Disp U
```

**PROGRAM:SUITES3**

```
: 1 → P
: 2 → U
: EffListe L1
: Prompt N
: While P ≤ N
: 2U + 5 → U
: U → L1(P)
: P + 1 → P
: End
```

1. Ces trois programmes permettent de calculer les termes d'une suite définie par récurrence. Laquelle ?

2. Saisir et exécuter les trois programmes dans la machine. Quelle est la spécificité de chacun ?

3. Quel programme est le plus adapté à chaque situation ?

« Calculer tous les termes de  $u_1$  à  $u_{20}$  »

« Calculer  $u_{20}$  »

« Calculer  $u_6$ ,  $u_{13}$ ,  $u_{17}$  et  $u_{20}$  »

## CORRIGE – NOTRE DAME DE LA MERCI - MONTPELLIER

## EXERCICE 3C.1

Dans chaque cas, indiquer les valeurs successives des variables pendant l'exécution de l'algorithme, et entourer les valeurs qui s'affichent à l'écran :

1.

*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U
0	3	2
1	3	5
2	3	8
3	3	11
4	3	11

2.

*U prend la valeur 0*  
*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Afficher *U*  
 Fin de boucle

P	N	U
		0
1	3	2
2	3	8
3	3	26
4	3	

3.

*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U
1	3	5
2	3	8
3	3	11
4	3	11

4.

*U prend la valeur 0*  
*P prend la valeur 1*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U
		0
1	3	2
2	3	8
3	3	26
4	3	26

5.

*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*P + 2*  
   *P prend la valeur P+1*  
 Afficher *U*  
 Fin de boucle

P	N	U
0	3	2
1	3	5
2	3	8
3	3	11
4	3	

6.

*U prend la valeur 1*  
*P prend la valeur 0*  
*N prend la valeur 3*  
 Tant que *P* est inférieur ou égal à *N* :  
   *U prend la valeur 3\*U + 2*  
   *P prend la valeur P+1*  
 Fin de boucle  
 Afficher *U*

P	N	U
		1
0	3	5
1	3	17
2	3	83
3	3	251
4	3	251

b. Parmi tous ces algorithmes, indiquer celui qui permet de calculer le plus rapidement possible  $u_3$ , où  $u_n$  est la suite

définie par récurrence par : 
$$\begin{cases} u_0 = 0 \\ u_{n+1} = 3u_n + 2 \end{cases} \dots \rightarrow \text{le } 4^{\text{ème}} \text{ algorithme.}$$

## EXERCICE 3C.2

On considère les programmes (TI-82) suivants

PROGRAM:SUITES1

```
: 1→P
: 2→U
: Prompt N
: While P≤N
: 2U+5→U
: P+1→P
: Disp U
: End
```

PROGRAM:SUITES2

```
: 1→P
: 2→U
: Prompt N
: While P≤N
: 2U+5→U
: P+1→P
: End
: Disp U
```

PROGRAM:SUITES3

```
: 1→P
: 2→U
: EffListe L1
: Prompt N
: While P≤N
: 2U+5→U
: U→L1(P)
: P+1→P
: End
```

1. Ces trois programmes permettent de calculer les termes d'une suite définie par récurrence. Laquelle ?

$$\begin{cases} u_0 = 2 \\ u_{n+1} = 2u_n + 5 \end{cases}$$

2. Saisir et exécuter les trois programmes dans la machine. Quelle est la spécificité de chacun ?

Le premier programme affiche toutes les valeurs de la suite jusqu'au rang demandé.

Le deuxième programme affiche directement la valeur de la suite au rang demandé.

Le troisième programme stocke dans une table toutes les valeurs de la suite jusqu'au rang demandé.

3. Quel programme est le plus adapté à chaque situation ?

« Calculer tous les termes de  $u_1$  à  $u_{20}$  » → le 1<sup>er</sup> ou le 3<sup>ème</sup> programme

« Calculer  $u_{20}$  » → le 2<sup>ème</sup> programme

« Calculer  $u_6$ ,  $u_{13}$ ,  $u_{17}$  et  $u_{20}$  » → le 1<sup>er</sup> ou le 3<sup>ème</sup> programme

