

1) La liste « résultats » lorsque $n = 4$

Si l'on suit pas à pas l'exécution du programme, on obtient :

- Initialisation du programme

La liste « résultats » est vidée.

Si l'utilisateur choisit le nombre 4, on aura $n = 4$ et, pour commencer, $i = 1$

Le programme de la boucle qui suit doit être réalisé 4 fois.

- Boucle 1 :

Puisque « 4 modulo 1 » renvoie 0, alors 1 est mis dans la liste « résultats ».

Puis la variable i augmente de 1 donc i prend la valeur 2.

- Boucle 2 :

« 4 modulo 2 » renvoie 0, donc 2 est ajouté à la liste « résultats » puis i prend la valeur 3.

- Boucle 3 :

« 4 modulo 3 » renvoie 1, donc 3 n'est pas ajouté à la liste « résultats » et i prend la valeur 4.

- Boucle 4:

« 4 modulo 4 » renvoie 0, donc 4 est ajouté à la liste « résultats » puis i prend la valeur 5.

- Fin du programme

La liste « résultats » s'affiche : (1 ; 2 ; 4).

Si l'utilisateur choisit le nombre 4, la liste « résultats » sera (1 ; 2 ; 4).

Remarque

Pour bien comprendre l'évolution de l'état des variables et de la liste, on peut détailler davantage l'exécution du programme (voir tableau suivant), en numérotant de 1 à 9 chacune des lignes de ce script (la ligne 7.3, par exemple, correspondant dans la boucle à la 3^{ème} exécution de la ligne 7).

		Valeur de n	Valeur de i	Réponse à l'instruction conditionnelle « si $n \text{ modulo } i = 0$ »	Contenu de la liste résultats
Avant la boucle	ligne 2	-	-	-	vidée
	ligne 3	-	-	-	$\emptyset$
	ligne 4	4	-	-	$\emptyset$
	ligne 5	4	1	-	$\emptyset$
1 ^{ère} exécution de la boucle	ligne 7.1	4	1	OUI	$\emptyset$
	ligne 8.1	4	1	-	(1)
	ligne 9.1	4	2	-	(1)
2 ^{ème} exécution de la boucle	ligne 7.2	4	2	OUI	(1)
	ligne 8.2	4	2	-	(1 ; 2)
	ligne 9.2	4	3	-	(1 ; 2)
3 ^{ème} exécution de la boucle	ligne 7.3	4	3	NON	(1 ; 2)
	ligne 8.3	4	3	-	(1 ; 2)
	ligne 9.3	4	4	-	(1 ; 2)
4 ^{ème} exécution de la boucle	ligne 7.4	4	4	OUI	(1 ; 2)
	ligne 8.4	4	4	-	(1 ; 2 ; 4)
	ligne 9.4	4	4	-	(1 ; 2 ; 4)

2) a) Contenu de la liste une fois le programme exécuté

Ce script définit une liste nommée « résultats » qui est vidée de son contenu en début d'exécution du programme (ligne 2).

Ce script définit en lignes 4 et 5 deux variables :

- la variable n est le nombre choisi par l'utilisateur. Cette variable reste constante tout au long de l'exécution du script ;
- la variable i prend pour valeur initiale 1 et est ensuite incrémentée de 1 à chacune des n itérations de la boucle. La variable i prend donc successivement toutes les valeurs entières de 1 à n .

Ce script comporte une boucle répétée n fois.

La condition « si $n \text{ modulo } i = 0$ » de la ligne 7 peut se reformuler en « si i divise n ».

Ainsi en ligne 8, un nombre i vient s'ajouter à la liste « résultats » uniquement s'il est un diviseur de n .

Comme i prend successivement toutes les valeurs de 1 à n , à la fin de l'exécution du programme la liste « résultats » est constituée de tous les diviseurs du nombre n choisi, diviseurs qui sont listés dans l'ordre croissant.

Ainsi, la liste « résultats » est constituée de tous les diviseurs de n présentés dans l'ordre croissant.

Remarque

Le nombre entier n choisi ne doit pas être nul pour que l'instruction « répéter n fois » puisse être exécutée. Sinon, si l'on choisit le nombre 0, la liste « résultats » restera vide.

2) b) Nombre choisi si la liste « résultats » ne contient qu'un seul nombre

La question posée revient à déterminer les nombres entiers ne possédant qu'un seul diviseur.

On sait que tout nombre entier naturel non nul autre que 1 admet au moins deux diviseurs, 1 et lui-même.

1 est le seul nombre entier naturel non nul à ne posséder qu'un seul diviseur, lui-même.

Ainsi, si la liste « résultats » ne contient qu'un seul nombre une fois le programme exécuté, c'est que le nombre choisi était 1.

2) c) Nombres choisis si la liste « résultats » contient exactement deux nombres

La question posée revient à déterminer les nombres entiers possédant exactement deux diviseurs.

On sait que tout entier naturel non nul autre que 1 possède au moins deux diviseurs (1 et lui-même) et que les nombres qui ont exactement deux diviseurs sont, par définition, les nombres premiers.

Donc si la liste « *résultat* » contient exactement deux nombres, c'est que le nombre choisi est un nombre premier.