UNE PROPRIETE DES ENTIERS NATURELS.

Voici un petit algorithme qui montre une propriété des nombres naturels.

|  |
| --- |
| ***Demander et lire un nombre N entier positif non nul Recommencer l'action suivante:     \* si N est pair, il est divisé par 2     \* si N est impair, il est multiplié par 3 et augmenté de 1 Arrêter quand N vaut 1.*** |

***Exemples:***  
N = 26 devient 13, 40, 20, 10, 5, 16, 8, 4, 2, 1  
N = 7 devient 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

***Note:***

D’après une source trouvée sur Internet, il a été vérifié par ordinateur que tout nombre entier compris entre 1 et 1.099.511.627.776 était ramené à 1 après un nombre fini d'étapes. Il s’agirait d’une propriété qui n’aurait pas encore été démontrée dans le cas général. On attend toujours la découverte d'un nombre pour lequel cette propriété ne serait pas vraie.

***Ecrire un programme permettant d’utiliser cet algorithme, avec ALGOBOX.***

**Solution :**

1° Déclarer les variables : N ( l’entier sur lequel on veut travailler),

A ( une des valeurs que prend l’entier N lors d’une étape, au cours de ses transformations successives),

I ( compte le nombre d’opérations subies par N pour arriver finalement à 1) ,

R ( reste de la division de A par 2, qui permet de tester si A est pair ou impair),

T : test (initialisé à 0, on peut lui affecter par exemple la valeur 1 lorsque A = 1. On continue tant que A = 0).

2° Initialiser T ( test) et I ( compteur) à 0 puis lire l’entier N. Prévoir de s’arrêter si N=1 en donnant alors à T la valeur 1. Attention : instruction == pour tous les tests…

3° Tant que T prend la valeur 0 faire : A prend la valeur N ; R prend la valeur A%2 ; Si R est nul, A prend la valeur N/2 ; sinon A prend la valeur 3\*N+1 ; faire afficher A (étape intermédiaire) avec un peu d’espace (pour une bonne lisibilité des résultats).

Si A = 1 faire T= 1 ( arrête le processus). Puis I prend la valeur I +1 et N prend la valeur A pour continuer le processus « tant que T=0 »

4° Il reste à afficher le nombre d’opérations pour arriver à 1 (par commodité, et pour une meilleure lisibilité ,afficher avant cela un espace avec passage à la ligne)

On obtient :

|  |  |
| --- | --- |
| 1 VARIABLES  2 N EST\_DU\_TYPE NOMBRE  3 A EST\_DU\_TYPE NOMBRE  4 I EST\_ DU\_TYPE NOMBRE  5 R EST\_DU\_TYPE NOMBRE  6 T EST\_DU\_TYPE NOMBRE  7 DEBUT\_ALGORITHME  8 T PREND\_LA\_VALEUR 0  9 I PREND\_LA\_VALEUR 0  10 LIRE N  11 SI (N==1) ALORS  12 DEBUT\_SI  13 T PREND\_LA\_VALEUR 1  14 FIN\_SI  15 TANT\_QUE (T==0) FAIRE  16 DEBUT\_TANT\_QUE  17 A PREND\_LA\_VALEUR N  18 R PREND\_LA\_VALEUR A%2 | 19 SI (R==0) ALORS  20 DEBUT\_SI  21 A PREND\_LA\_VALEUR N/2  22 FIN\_SI  23 SINON  24 DEBUT\_SINON  25 A PREND\_LA\_VALEUR 3\*N+1  26 FIN\_SINON  27 AFFICHER A  28 AFFICHER " "  29 SI (A==1) ALORS  30 DEBUT\_SI  31 T PREND\_LA\_VALEUR 1  32 FIN\_SI  33 I PREND\_LA\_VALEUR I+1  34 N PREND\_LA\_VALEUR A  35 FIN\_TANT\_QUE  36 AFFICHER " "  37 AFFICHER "Nombre d'opérations: "  38 AFFICHER I  39 FIN\_ALGORITHME |

**Tests :**

|  |  |
| --- | --- |
| **Exemple avec N = 7 :**  \*\*\*Algorithme lancé\*\*\*  22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1  Nombre d'opérations: 16  \*\*\*Algorithme terminé\*\*\* | **Exemple avec N=26 :**  \*\*\*Algorithme lancé\*\*\*  13 40 20 10 5 16 8 4 2 1  Nombre d'opérations: 10  \*\*\*Algorithme terminé\*\*\* |
| **Exemple avec N=31 :**  \*\*\*Algorithme lancé\*\*\*  94 47 142 71 214 107 322 161 484 242 121 364 182 91 274 137 412 206 103 310 155 466 233 700 350 175 526 263 790 395 1186 593 1780 890 445 1336 668 334 167 502 251 754 377 1132 566 283 850 425 1276 638 319 958 479 1438 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367 4102 2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 488 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1  Nombre d'opérations: 106  \*\*\*Algorithme terminé\*\*\* | |