|  |  |
| --- | --- |
| Ponctuel 10 | Les files (algorithmes) |

# 

1. Dites ce qui sera affiché après l’exécution des algorithmes suivants :

|  |  |
| --- | --- |
| a)  file.initialiser ( )  file.enfiler ( 100 );  pour k🡨 1à 5  file.enfiler ( 100 + k );  fin pour  x 🡨 file.defiler ( )  y 🡨 file.defiler ( )  file.enfiler ( 200 );  afficher "x vaut ", x , " et y vaut " , y  tant que non file.vide ( )  afficher file.defiler ( )  fin tant que | b)  file1.initialiser ( ) file2.initialiser ( ) pour k🡨 1 à 10 faire  file1.enfiler ( k )  fin pour tant que non file1.vide ( )   element 🡨 file1.defiler ( )  si (element mod 2 = 0) alors  file2.enfiler ( element )  sinon  file1.enfiler ( element \* 2 )  fin si  fin tant que tant que non file1.vide ( )   afficher file1.defiler ( )  fin tant que  tant que non file2.vide ( )   afficher file2.defiler ( )  fin tant que |

|  |  |
| --- | --- |
| a) | b) |

1. On désire écrire un algorithme d’une application qui **retire** de la file d’entiers *file* tous les éléments plus grands que 100. Commentez chacune des solutions suivantes :

|  |  |
| --- | --- |
| *Version a)*  tant que non file.vide ( )  element 🡨 file.defiler ( )  si element <= 100 alors  file.enfiler ( element )  fin si  fin tant que | *Version b)*  file2.initialiser ( )  tant que non file.vide ( )  element 🡨 file.defiler ( )  si element <= 100 alors  file2.enfiler ( element )  fin si  fin tant que  tant que non file2.vide ( )  file.enfiler ( file2.defiler ( ) )  fin tant que |

|  |  |
| --- | --- |
| a) | b) |

1. Écrire un algorithme qui permet d’afficher le nombre d’éléments dans la file *file.* La filedoit évidemment être intacte à la fin du traitement.