|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 2h | **Structures bouclées** |  |
| Objectif : savoir réaliser des instructions de répétitions. | | |
| **Matériel :** python | | |

|  |  |
| --- | --- |
| **Structure :** | **Algorigramme :** |
| Linéaire :  Suite d'actions à exécuter successivement dans l'ordre. |  |
| Conditionnelle (ou alternative) :  Test offrant 2 choix possibles s'excluant mutuellement. |  |
| Répétitive (ou itérative) :  **Boucle non bornée avec pré-test :**  On test la condition, tant qu'elle est vraie alors on réalise les actions. |  |
| Répétitive (ou itérative) :  **Boucle bornée :**  Initialisation du compteur.  Les actions sont répétées un nombre (compteur) de fois prédéfinies.  Dans la boucle une opération agit sur le compteur. |  |

* Créez un programme qui réalise les tâches suivantes :
* Le PC choisira un nombre entier aléatoirement entre 1 et 100.
* Le PC demandera à l’utilisateur de deviner ce nombre (1 essai seulement).
  + Si l’utilisateur trouve le nombre, **"Bravo"** sera affiché sinon le programme affichera "**C'est plus"** ou **"C'est moins"** selon le cas.

Pour réaliser les programmes, on utilisera les variables suivantes :

|  |  |  |
| --- | --- | --- |
| **Nom de la variable :** | **Rôle de la variable :** | **Type de la variable :** |
| nb\_ordi | Nombre aléatoire | entier |
| nb\_utilisateur | Nombre proposé | entier |
| nb\_essais | Comptage des essais | entier |

**Aide :** la méthode randint(1,n) renvoie un nombre entier choisi aléatoirement entre 1 et n.

|  |
| --- |
| from random import randint  nombre = randint(1,100) |

→ Testez le programme.

* Modifiez votre programme de sorte qu’il réalise la tâche suivante :
* Répétition du programme précédent 10 fois, quelque soit le résultat, sauf pour le nombre aléatoire qui ne sera demandé qu'une fois.

**Aide :** la boucle bornée.

|  |  |  |
| --- | --- | --- |
| **Algorithme :** | **Programmation Python :** | **Exemple Python :** |
| Pour i allant de 0 à n-1 faire  Instruction1  Instruction2  Fin de pour | For i in range(n) :  Instruction1  Instruction2 | Ce programme affichera 100 fois "Je n'embêterai plus ma petite soeur". |

**Remarque :** for i in range(100) = for i in range(0,100,1)

→ Testez le programme.

**Pour votre programme la boucle "for" est une mauvaise solution de répétition car même si l'utilisateur trouve le bon nombre, le programme s'exécutera 10 fois → boucle "while".**

* Modifiez votre programme de sorte qu’il réalise les tâches suivantes :
* Le PC choisira un nombre entier aléatoirement entre 1 et 100.
* Le PC demandera à l’utilisateur de deviner ce nombre.
* Tant que l’utilisateur ne trouve pas ce nombre, il est invité à recommencer et un message "**C'est plus"** ou **"C'est moins"** est affiché.
* Lorsque l'utilisateur trouvera le bon nombre le programme affichera **"Bravo"**.

**Aide :** la boucle non bornée.

|  |  |  |
| --- | --- | --- |
| **Algorithme :** | **Programmation Python :** | **Exemple Python :** |
| Initialiser condition  Tant que condition vraie faire  Instruction1  Instruction2  Modifier condition  Fin de tant que | ....  while condition vraie :  Instruction1  Instruction2  .... | Tant que la réponse ne sera pas celle attendue, ce programme reposera la même question. |

→ Testez le programme.

* Modifiez votre programme de sorte qu’il réalise les tâches suivantes :
* Arrêt du programme au bout de 10 essais maximum**.**
* Si l'utilisateur trouve le nombre en moins de 10 essais, le programme affichera le nombre d'essais réalisé et affichera **"Bravo"**.
* Si l'utilisateur ne trouve pas le nombre au bout de 10 essais le programme affichera le nombre qu'il fallait trouver.

**Aide :** while avec plusieurs conditions

|  |
| --- |
| while condition1 vraie **and** condition2 vraie :  instruction |

→ Testez le programme.

→ Rédigez des commentaires dans votre programme.

→ Sur feuille, écrivez l'algorithme du programme final avec des mots en Français.