

Les sous-programmes

Exercices

Tir à l'arc - Bac TP 2019

Le jeu de tir à l'arc consiste à tirer des flèches en utilisant un arc vers un endroit cible.

Un club de tir à l'arc organise une compétition entre ses joueurs. Cette compétition consiste à donner à chaque joueur trois essais de tir.

A chaque tir, le joueur gagne de 0 à 10 points selon l'endroit touché par la flèche sur une cible composée de 10 cercles concentriques comme l'illustre la figure ci-contre.



Après le 3ème essai, un score est calculé pour chaque joueur représentant la somme des résultats de ses trois essais, et ce pour classer les joueurs.

On se propose d'écrire un programme permettant :

- de remplir un tableau A par N noms de joueurs participant à cette compétition (avec $2 \leq N \leq 20$),
- de calculer le score de chaque joueur sachant que le résultat d'un essai est un entier positif inférieur ou égal à 10,
- d'afficher la liste des joueurs ainsi que les scores correspondants en commençant par le meilleur.

Exemple : Pour $N = 4$ et le tableau A suivant :

A	Mohamed ABIDI	Fatma AYADI	Anis OMRI	Imen LOUIZ
	1	2	3	4

En considérant que les résultats des trois essais ainsi que les scores calculés des quatre joueurs sont les suivants :

Joueur	Essai n°1	Essai n°2	Essai n°3	Score
Mohamed ABIDI	8	10	9	27
Fatma AYADI	10	7	10	27
Anis OMRI	7	9	7	23
Imen LOUIZ	9	9	10	28

La liste des joueurs affichée sera la suivante :

- Imen LOUIZ avec un score de 28
- Mohamed ABIDI avec un score de 27
- Fatma AYADI avec un score de 27
- Anis OMRI avec un score de 23

Ci-après, un algorithme du programme principal dans lequel des modifications seront apportées pour répondre aux besoins cités ci-dessus :

Algorithme

Algorithme TirArc

Début

// Séquence 1

Répéter

Ecrire ("Donner le nombre de joueurs : "); Lire (N)

Jusqu'à (N dans [2..20])

// Séquence 2

Pour i de 1 à N **faire****Répéter**

Ecrire ("Donner le nom du joueur n°", i, " : "), Lire (A[i])

Jusqu'à (alpha (A[i]))

Fin pour

// Séquence 3

Score (A, N)

Fin

Travail demandé :

1. Implémenter l'algorithme TirArc en Python et ajouter les déclarations nécessaires puis enregistrer le programme, dans votre dossier de travail, sous le nom "TirArc0".

NB : Il est recommandé de limiter la taille des éléments du tableau A à 30 caractères.

2. Dans votre dossier de travail, créer une deuxième copie de ce programme et l'enregistrer sous le nom "TirArc1".

Dans ce fichier :

- a. Transformer la séquence 2 en un module et apporter les modifications nécessaires dans le programme principal.
- b. Développer la fonction Alpha qui permet de vérifier si le nom d'un joueur est une suite de 30 lettres au maximum et pouvant contenir des espaces.
- c. Développer le module Score permettant :
 - de saisir les résultats des trois essais de chaque joueur et de calculer son score,
 - d'afficher la liste des joueurs ainsi que leurs scores selon un ordre décroissant des scores.

Plagiarisme - Bac TP 2019

Le plagiat est une mauvaise pratique qui consiste à copier ou s'inspirer fortement d'une œuvre sans citer son auteur et en faisant passer cette œuvre pour la sienne.

On suppose qu'un texte est plagié lorsque le nombre de mots communs dépasse la moitié du nombre de mots du texte d'origine.

Exemple

Texte à tester : "ENTRE DEUX VERSIONS DU MEME DOCUMENT CE LOGICIEL PERMET DE CHERCHER LE NOMBRE DE MOTS IDENTIQUES."

Texte d'origine : "CE PROGRAMME PERMET DE TROUVER LE NOMBRE DE MOTS IDENTIQUES DANS DEUX VERSIONS DU MEME TEXTE." "

Les mots communs aux deux textes sont :

DEUX – VERSIONS – DU – MEME – CE – PERMET – DE – LE – NOMBRE – DE – MOTS – IDENTIQUES

Le texte testé est plagié. En effet, le nombre de mots communs aux deux textes est égal à 12, dépassant ainsi la moitié du nombre de mots du texte d'origine qui est 16.

On se propose d'écrire un programme qui permet de saisir deux textes T1 et T2 ayant un même nombre de mots, de vérifier si T2 est plagiat de T1 selon le principe décrit précédemment et d'afficher un message correspondant.

Ci-après, un algorithme du programme principal dans lequel des modifications seront apportées pour répondre aux besoins cités ci-dessus, avec Nbc désigne le nombre de mots communs à T1 et T2 :

Algorithme

```

Algorithme Plagiat
Début
  // Séquence 1
  Répéter
    Lire (T1)
    Lire (T2)
  Jusqu'à (Nb mots (T1) = Nb mots (T2))
  // Séquence 2
  Nbc ← Commun (T1, T2)
Fin

```

Travail demandé

1. Implémenter l'algorithme Plagiat en Python et ajouter les déclarations nécessaires puis enregistrer le programme, dans votre dossier de travail, sous le nom "Plagiat0".
2. Dans votre dossier de travail, créer une deuxième copie de ce programme et l'enregistrer sous le nom "Plagiat1". Dans ce fichier :
 - a. Développer un module intitulé Saisir qui permet de saisir une chaîne commençant obligatoirement par une lettre, composée au maximum de 200 caractères, se terminant par un point et ne pouvant contenir que des lettres majuscules et des espaces. Puis, apporter les modifications nécessaires dans le programme principal. Ce module permet aussi d'enlever les espaces superflus (garder un seul espace entre deux mots consécutifs et ne garder aucun espace avant le point de fin de texte).
 - b. Développer le module Nb mots qui permet de déterminer le nombre de mots dans un texte.
 - c. Développer le module Commun, qui permet de déterminer le nombre de mots communs dans deux textes donnés.
 - d. Apporter les modifications nécessaires au programme afin de vérifier si le texte testé est plagié et d'afficher le message correspondant.

Fidélité - Bac TP 2019

Un gérant d'une salle de sport veut récompenser les adhérents fidèles en leur offrant un bonus sous forme d'heures d'entraînement gratuites, calculé à partir de leurs numéros d'abonnement.

Le bonus est calculé en fonction de l'ancienneté de l'adhérent, exprimée en nombre de mois par rapport à la date du jour. En effet, une heure supplémentaire est offerte pour chaque mois d'ancienneté, sachant que le bonus ne sera pris en considération que si l'ancienneté dépasse 5 ans.

Un numéro d'abonnement est formé de 10 caractères dont le premier est une lettre majuscule indiquant la catégorie de l'adhérent (A : Adulte, J : Junior, E : Enfant) suivi de 9 chiffres répartis comme suit :

1. Les 4 premiers chiffres représentent l'année d'adhésion qui doit être comprise entre 2000 et 2019.
2. Les 2 suivants représentent le mois d'adhésion dont la valeur doit être comprise entre 1 et 12.
3. Les 3 derniers chiffres représentent le numéro d'adhésion qu'on suppose distinct pour tous les adhérents.

Exemple

Pour le numéro d'abonnement E201403080, l'adhérent est un enfant dont l'année d'adhésion est 2014, le mois d'adhésion est 03 (mars) et son numéro d'adhésion est 080. Le bonus accordé à cet adhérent est de 62 heures. En effet, son ancienneté est égale à cinq ans et deux mois par rapport à la date d'aujourd'hui (28/05/2019), en nombre de mois elle est égale à 62 ($12 \times 5 + 2$).

On se propose d'écrire un programme qui permet :

- de saisir un entier N représentant le nombre d'adhérents de cette salle (avec $5 \leq N \leq 30$),
- de remplir un tableau TA par les N numéros d'abonnements de ces adhérents en respectant le format décrit précédemment,
- d'afficher les numéros d'abonnements des adhérents ayant une ancienneté supérieure à 5 ans et appartenant à une catégorie donnée ainsi que leurs bonus correspondants sachant que la date du jour à considérer est 28/05/2019.

Ci-après, un algorithme du programme principal dans lequel des modifications seront apportées pour répondre aux besoins cités ci-dessus :

Algorithme

Algorithme Fidelite

Début

// Séquence 1

Répéter

Ecrire ("Donner le nombre des adhérents :"); Lire (N)

Jusqu'à (N dans [5..30])

// Séquence 2

Remplir (TA, N)

Bonus (TA, N)

Fin

Travail demandé

1. Implémenter l'algorithme Fidelite en Python et ajouter les déclarations nécessaires puis enregistrer le programme, dans votre dossier de travail, sous le nom "Fidele0".
NB : Il est recommandé de limiter la taille des éléments du tableau TA à 10 caractères.
2. Dans votre dossier de travail, créer une deuxième copie de ce programme et l'enregistrer sous le nom "Fidele1".
Dans ce fichier :
 - a. Transformer la séquence 1 en un module et apporter les modifications nécessaires dans le programme principal.
 - b. Développer le module Remplir qui permet de remplir un tableau TA par les N numéros d'abonnements en respectant la forme décrite précédemment.
 - c. Développer le module Bonus permettant :
 - de saisir une catégorie d'adhérents, qui doit être une lettre majuscule parmi les suivantes : A, J ou E,
 - d'afficher les numéros ainsi que les bonus correspondants aux abonnements appartenant à la catégorie donnée et ayant une ancienneté supérieure à 5 ans.

Heure Miroir - BAC TP 2019

Un spécialiste en psychologie analytique veut faire une étude sur les heures qui se disent heures miroirs et ce, à partir des heures de naissances des nouveau-nés. Les heures miroirs sont classées selon trois types à savoir les heures miroirs doublées, les heures miroirs triplées et les heures miroirs inversées.

Pour commencer cette étude, il a demandé auprès de l'administration d'un hôpital de lui fournir, pour une journée donnée, la liste des noms et prénoms des nouveau-nés ainsi que les heures de leurs naissances.

Selon la représentation horaire HH:MM, l'heure miroir est dite :

- doublée si le nombre qui représente l'heure est le même que celui qui représente les minutes (exemples 13:13, 22:22, 15:15, ...)
- triplée s'il existe une séquence de trois chiffres identiques et consécutifs (exemples 22:22, 22:29, 10:00, ...)
- inversée si les chiffres qui représentent les minutes sont l'inverse de ceux qui représentent l'heure (exemples 13:31, 14:41, 22:22, ...)

NB : Une heure miroir peut être à la fois doublée, triplée et inversée.

Pour faire cette étude, on se propose d'écrire un programme qui permet de remplir deux tableaux NP et HN respectivement par :

- les noms et les prénoms des N nouveau-nés de la journée,
- l'heure de naissance de chacun,

puis d'afficher :

- la liste des nouveau-nés ayant une heure de naissance miroir, en précisant pour chacun le type de l'heure miroir correspondant (doublée, triplée ou inversée),
- le nombre des nouveau-nés ayant l'heure de naissance miroir doublée,
- le nombre des nouveau-nés ayant l'heure de naissance miroir triplée,
- le nombre des nouveau-nés ayant l'heure de naissance miroir inversée.

Exemple :

Pour N = 10 et les tableaux NP et HN suivants :

NP	Rania Kefi	Karim Salhi	Hatem Krid	Sahar Feki	Maha Sessi	Fedi Krayem	Amin Douaji	Dina Selmi	Samir Siala	Abir Zekri
HN	06:05	11:14	12:23	13:13	14:44	15:51	20:33	21:11	22:22	23:07

Le programme affiche :

Les nouveau-nés ayant l'heure de naissance miroir sont :

Karim Salhi : heure miroir triplée
 Sahar Feki : heure miroir doublée
 Maha Sessi : heure miroir triplée
 Fedi Krayem : heure miroir inversée
 Dina Selmi : heure miroir triplée
 Samir Siala : heure miroir doublée
 Samir Siala : heure miroir triplée
 Samir Siala : heure miroir inversée

Le nombre des nouveau-nés ayant l'heure de naissance miroir doublée est 2.

Le nombre des nouveau-nés ayant l'heure de naissance miroir triplée est 4.

Le nombre des nouveau-nés ayant l'heure de naissance miroir inversée est 2.

Ci-après, un algorithme du programme principal dans lequel des modifications seront apportées pour répondre aux besoins cités ci-dessus :

Algorithme

Algorithme HeureMiroir

Début

// Séquence 1

Ecrire ("Donner le nombre de nouveau-nés : "); Lire (N)

// Séquence 2

Remplissage (NP, HN, N)

Affichage (NP, HN, N)

Fin

Travail demandé

1. Implémenter l'algorithme HeureMiroir en Python et ajouter les déclarations nécessaires puis enregistrer le programme, dans votre dossier de travail, sous le nom "HMiroir0".
NB : Il est recommandé de limiter la taille des éléments des tableaux à 30 caractères.
2. Dans votre dossier de travail, créer une deuxième copie de ce programme et l'enregistrer sous le nom "HMiroir1".
Dans ce fichier :
 - a. Transformer la séquence 1 en un module en ajoutant les contrôles de saisie nécessaires pour que N soit un entier de l'intervalle [3,20] et apporter les modifications adéquates dans le programme principal.
 - b. Développer le module Remplissage permettant de remplir :
 - le tableau NP par les noms et les prénoms des N nouveau-nés de la journée, sachant que le nom et le prénom est une seule chaîne commençant par une lettre majuscule et formée au maximum de 30 caractères,
 - le tableau HN par l'heure de naissance de chaque nouveau-né, et ce en respectant le format horaire HH:MM, sachant que la valeur de HH est comprise entre 0 et 23 et celle de MM est comprise entre 0 et 59 représentée chacune sur deux chiffres.
 - c. Développer le module Affichage permettant de faire les affichages nécessaires comme indiqué précédemment.

NB : Le candidat n'est pas appelé à respecter l'ordre d'affichage de l'exemple.

Collection - BAC TP 2018

Un client collectionne les images qu'il trouve dans les paquets de ses biscuits préférés pour gagner des passes d'entrées gratuites, à un parc d'attraction. La collection doit comporter 20 images différentes relatives à des personnages distincts de bandes dessinées.

On suppose que les images ont été disposées au hasard dans les paquets et que :

- chaque paquet contient exactement une image de personnages de bandes dessinées,
- un même personnage peut se retrouver dans plusieurs paquets.

Pour gagner ce voyage, le client achète une boîte contenant 50 paquets.

On donne l'algorithme du programme principal suivant où P représente un tableau contenant la collection des personnages distincts que le client a pu collectionner, suite à l'ouverture des 50 paquets achetés, et C, le nombre d'éléments de P.

Algorithme

Algorithme Collection

Début

Répéter // Séquence 1

Ecrire ("Donner le nom du personnage trouvé dans le paquet n° 1 :") ; Lire (P[1])

Jusqu'à (FN Valide (P[1]))

C ← 1 // Séquence 2 et 3

Pour i de 2 à 50 faire

Répéter

Ecrire ("Donner le nom du personnage trouvé dans le paquet n° ",i, ":") ; Lire (Pers)

Jusqu'à (FN Valide (Pers))

Si (Chercher (Pers, P,C)=faux) Alors

C ← C+1

P[C] ← Pers

Fin Si

Fin Pour

// Séquence 4

Si (C = 20) Alors Ecrire ("Vous avez gagné") Sinon Ecrire ("Vous avez perdu") Fin Si

Fin

Travail à faire :

1. Implémenter l'algorithme Collection en Python et ajouter les déclarations nécessaires.
2. Développer la fonction Valide permettant de valider une chaîne qui doit être non vide, commencer par une majuscule et être formée uniquement de lettres non accentuées.
3. Développer la fonction Chercher qui permet de vérifier l'existence d'un personnage Pers dans le tableau P.
4. Mettre les séquences 2), 3) et 4) sous forme de commentaire et réécrire ces trois séquences en apportant les modifications nécessaires de façon à ce que le programme :
 - prenne en considération la possibilité de trouver 20 personnages distincts avant même de terminer l'ouverture des 50 paquets,
 - dans le cas, où le client n'a pas réussi à collectionner 20 personnages, affiche le nombre de personnages manquants pour gagner le concours.