

Programmation
Licence 1 UPEC 2021/2022
Travaux Machine 6 bis : Fonctions et *Syntax Error*

Dans ce TP on se familiarise avec la plateforme Web "Syntax Error"
<http://syntaxerror.lacl.fr>. Votre identifiant et mot de passe sont a priori les mêmes que sur Eprel. Il n'y a pas de limite de nombre de rendus, à chaque rendu votre programme est corrigé, et un message est affiché. Votre chargé de TM vous aidera dans l'utilisation.

Première connexion : à votre première connexion il est possible que votre compte ne soit pas reconnu, et que s'affiche l'erreur : "Utilisateur inconnu ou mauvais de mot de passe". Dans ce cas, allez sur la page <https://appetu.lacl.fr/profile/> puis cliquez sur "S'enregistrer" et authentifiez vous avec votre login et mot de passe de l'u-pec. Revenez sur le site <http://syntaxerror.lacl.fr>, votre identifiant devrait maintenant marcher. Une fois connecté vous devez vous inscrire au module "Programmation L1 S1 2021/2022 FST" puis cliquer sur "mettre à jour".

Vous devez soumettre un fichier .jvs pour chaque question.

Exercices

Exercice 1 : Fonctions simples Testez les fonctions suivantes sur le site "Syntax Error"

1. Écrire une fonction **somme** qui prend en paramètre un tableau **t** d'entiers et qui renvoie la somme de toutes les valeurs qui se trouvent dans le tableau.
2. Écrire une fonction **appartient** qui prend en paramètre un tableau **t** d'entiers et un entier **n** et qui renvoie **true** si **n** est dans une des cases de **t**, et **false** sinon.
3. Écrire une fonction **paire** qui prend en paramètre un tableau **t** d'entiers et qui renvoie **true** si **t** contient deux cases de suite qui ont une même valeur, et **false** sinon.
4. Écrire une fonction **brelan** qui prend en paramètre un tableau **t** d'entiers et qui renvoie **true** si **t** contient *trois* cases de suite qui ont une même valeur, et **false** sinon.

Exercice 2 : Operations Testez les fonctions suivantes sur le site "Syntax Error"

1. Écrire une fonction **sommeMultiple** qui prend en paramètre un tableau **t** d'entiers et un entier **n** et qui renvoie la somme de toutes les valeurs qui se trouvent dans une case dont l'indice est un multiple de **n**.
2. Écrire une fonction **tabMultiple** qui prend en paramètre un tableau **t** d'entiers et un entier **n** et qui renvoie un nouveau tableau qui contient seulement les valeurs qui se trouvent dans une case de **t** dont l'indice est un multiple de **n**.
3. Écrire une fonction **tabHasard** qui prend en paramètre deux entiers **n** et **h** et qui renvoie un tableau de taille **n** contenant **n** nombres différents choisis au hasard entre 0 et **h**.
4. Écrire une fonction **sommeHasard** qui prend en paramètre un tableau **t** d'entiers et un entier **n** et qui renvoie la somme de **n** entiers choisis au hasard dans **t**.

Exercice 3 : Modification d'un tableau Testez les fonctions suivantes sur le site "Syntax Error"

1. Écrire une fonction `void augmente2(int[] tab)` qui prend en entrée un tableau d'entiers, et qui augmente de 2 la valeur de chaque case du tableau.
2. Écrire une fonction `void augmente(int[] tab, int n)` qui prend en entrée un tableau d'entiers et un entier n , et qui augmente de n la valeur de chaque case du tableau.
3. Écrire une fonction `void echange(int[] tab, int i, int j)` qui prend en entrée un tableau d'entiers et deux entiers i, j , et qui échange de place les valeurs des cases i et j . Votre procédure ne doit pas vérifier si les entiers correspondent réellement à des cases du tableau.
4. Écrire une fonction `void rotation(int[] tab)` qui prend en entrée un tableau d'entiers et qui déplace chaque valeur d'une case à la suivante. La valeur de la dernière case sera mise dans la case 0.
5. Écrire une procédure `remplissage(int[] source1, int[] source2, int[] target)` qui prend en entrée trois tableaux d'entiers et qui commence à mettre les valeurs de `source1` dans `target` tant qu'il y a de la place. Ensuite elle met les valeurs de `source2` dans `target` tant qu'il y a de la place. S'il reste encore de la place dans `target`, on remplit les cases qui restent avec des 0. Testez votre procédure sur le site des tests.

Exercice 4 : Les permutations Testez ces fonctions sur les site "Syntax Error"

1. Écrire une fonction `contientApres`, qui prend en paramètre un tableau d'entiers `tab`, un entier `n`, et un entier `i` et renvoie `true` si le tableau contient `n` dans une case dont l'indice est plus grand ou égale à `i`.
2. Écrire une fonction `doublons`, qui prend en paramètre un tableau d'entier `tab` et renvoie `true` si le tableau contient un nombre plus qu'une fois. Cette fonction utilisera la fonction précédente.
3. Écrire une fonction `permutation`, qui prend en paramètre deux tableaux d'entiers et qui renvoie `true` si les deux tableaux ne contiennent pas de doublons et s'ils sont l'un une permutation de l'autre. On utilisera les deux fonctions précédentes.
4. Écrire une procédure `main` qui demande à l'utilisateur d'initialiser deux tableaux et qui affiche s'ils sont l'un permutation de l'autre (ce main ne sera pas testé sur Syntax Error).