

*17<sup>e</sup> édition*  
*30 mars 2023*



*L'arbre qui cache la forêt*

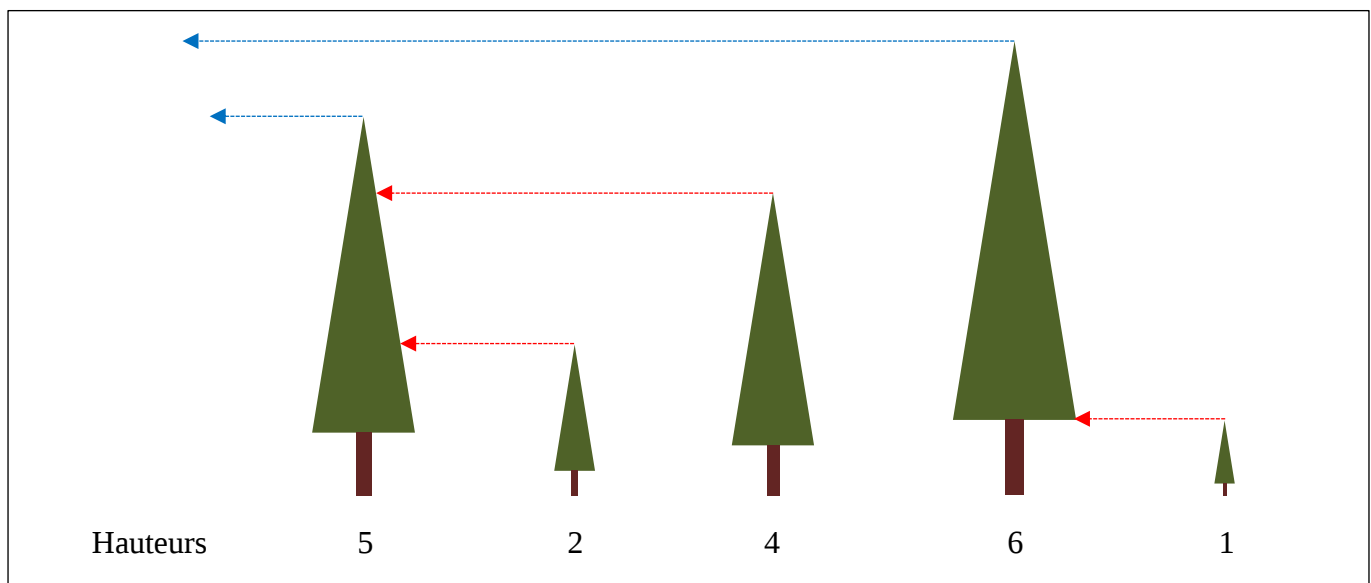
## Règlement

- Date et Lieu : La 17<sup>ème</sup> édition du concours de programmation de l'EPFC se déroulera le **jeudi 30 mars 2023 de 13h00 à 21h30** dans les locaux de l'**EPFC** (Avenue de l'Astronomie, 19 à 1210 Bruxelles – local C005). **Les candidats sont libres d'arriver et de repartir à l'heure qui leur convient.**
- Candidats : Le concours est ouvert aux étudiants en deuxième et troisième année du bachelier en informatique de l'**EPFC**, ainsi qu'aux anciens étudiants. Des dérogations pour d'autres étudiants peuvent être accordées. **Afin de faciliter l'organisation du concours, des préinscriptions auprès de A. SILOVY ([alsilovy@epfc.eu](mailto:alsilovy@epfc.eu)) sont souhaitées.**
- Epreuve : Programmation d'une (ou plusieurs) application(s) console. **Tout langage de programmation** approprié à l'écriture d'une telle application **est accepté**. En particulier, l'infrastructure de l'EPFC permet l'usage des langages Java, C/C++, C#, Python et Haskell. Les participants désireux d'utiliser un autre langage de programmation doivent s'adresser préalablement à l'un des responsables. A l'issue du concours, les candidats devront remettre le code source de leur solution (ainsi que l'exécutable) sur le serveur du réseau informatique de l'**EPFC**.
- Conditions : Les participants peuvent apporter toute documentation écrite ou électronique qu'ils peuvent juger utile. Ils peuvent donc employer des bibliothèques de code sur clé *USB*... Cependant, durant la durée de l'épreuve, ils ne peuvent pas communiquer avec l'"extérieur". L'usage d'*Internet* et du *GSM* (à l'exception évidente d'une impérieuse exigence) est proscrit.
- Classement : Contrairement aux habitudes pédagogiques, le premier critère de sélection sera le fonctionnement des exécutables répondants aux demandes formulées dans l'énoncé. Il correspond donc à **la correction du programme**. Pour départager les candidats, le classement se fera ensuite sur base des critères suivants :
- L'algorithme choisi (et sa complexité)
  - Le choix des structures de données
  - L'analyse du problème
  - L'élégance du code source
  - L'efficacité de la solution
- Prix : **Des prix sont prévus, soit sous forme de matériel, soit sous forme de bons d'achat.**
- Proclamation : La proclamation des résultats avec remise des prix sera organisée **le 6 avril 2023 à 18h**. Elle sera suivie d'un drink offert à tous les participants au concours

**Le problème** : Notre forêt est particulière : tous ses arbres sont alignés. Retrouvez, pour chaque arbre, celui qui lui cache la vue.

Un certain nombre d'arbres, de hauteurs diverses, sont alignés. Si l'on se place au sommet d'un arbre quelconque et que l'on regarde horizontalement vers l'arrière, en direction du début de la forêt, il est possible qu'un arbre plus haut que le nôtre nous masque la vue. Lequel est-ce ?

Par exemple, ces arbres de hauteurs respectives 5, 2, 4, 6 et 1 :



Bien sûr, rien ne masque la vue au premier arbre. La vue du deuxième arbre de hauteur 2 est masquée par le premier de hauteur 5. De même, la vue du troisième de hauteur 4 est masquée par le premier. En revanche, rien ne masque non plus la vue du quatrième arbre de hauteur 6...

Il vous est demandé de déterminer pour chaque arbre si sa vue est masquée, et si oui par quel autre arbre. Vous devez donner le numéro de l'arbre qui masque la vue ou zéro s'il n'y en a pas.

### Fonctionnement de votre programme

En pratique, votre programme lira sur l'entrée standard (au clavier) une première ligne contenant un nombre entier strictement positif  $N$  : le nombre d'arbres. Le programme lira ensuite une ligne avec  $N$  nombres entiers strictement positifs : les tailles des arbres.

Votre programme écrira sur la sortie standard (à l'écran)  $N$  entiers, séparés par des espaces, précisant pour chaque arbre le numéro du premier arbre dans le chemin qui lui masque la vue ou 0 s'il n'y en a pas.

## Consignes

Il est impératif que vous respectiez le format exact des Entrées/Sorties spécifié ci-dessus. Votre programme doit donc afficher **uniquement**  $N$  nombres entiers. En particulier, il ne doit pas afficher de message pour la saisie des données (du genre « entrez des nombres » ...) ni afficher un message du style « les numéros des arbres sont : ... ».

Cette consigne a pour objectif de permettre la mise en œuvre de tests automatiques de votre programme. Veillez à la respecter scrupuleusement. Votre programme pourrait être correct et néanmoins ne pas être validé pour cette raison.

Cependant, vous ne devez pas vérifier la saisie de l'utilisateur. Vous pouvez donc supposer que le format spécifié ci-dessus sera toujours respecté.

Par exemple, si votre programme se nomme `concours`, son exécution sur la ligne de commande pourrait donner lieu à l'interaction ci-dessous :

```
C:\>concours↵
5↵
5 2 4 6 1↵
0 1 1 0 4
C:\>
```

## Remarques

Dans cet exemple, les saisies de l'utilisateur sont soulignées. Le reste est affiché par l'ordinateur. Remarquez bien : aucune autre entrée/sortie que celles spécifiées plus haut. Quand le programme démarre, rien ne s'affiche. L'utilisateur entre 1 nombre entier sur la 1<sup>re</sup> ligne puis d'autres nombres entiers sur la 2<sup>e</sup>. Et le programme affiche  $N$  nombres entiers séparés par des espaces. **C'est tout !**

Des exemples de programmes, dans divers langages de programmation et respectant le même format d'entrées/sorties, vous sont fournis dans le dossier **DemosIO**.

Faites appel à un professeur pour vous aider si vous ne savez pas comment exécuter votre programme sur la ligne de commande.

## Réponses

Vous copierez vos solutions (sources et exécutables) dans un répertoire portant votre nom sur le serveur au point **PROFS-PUBLIC\CONCOURS\REPONSES**.

# Bon Amusement