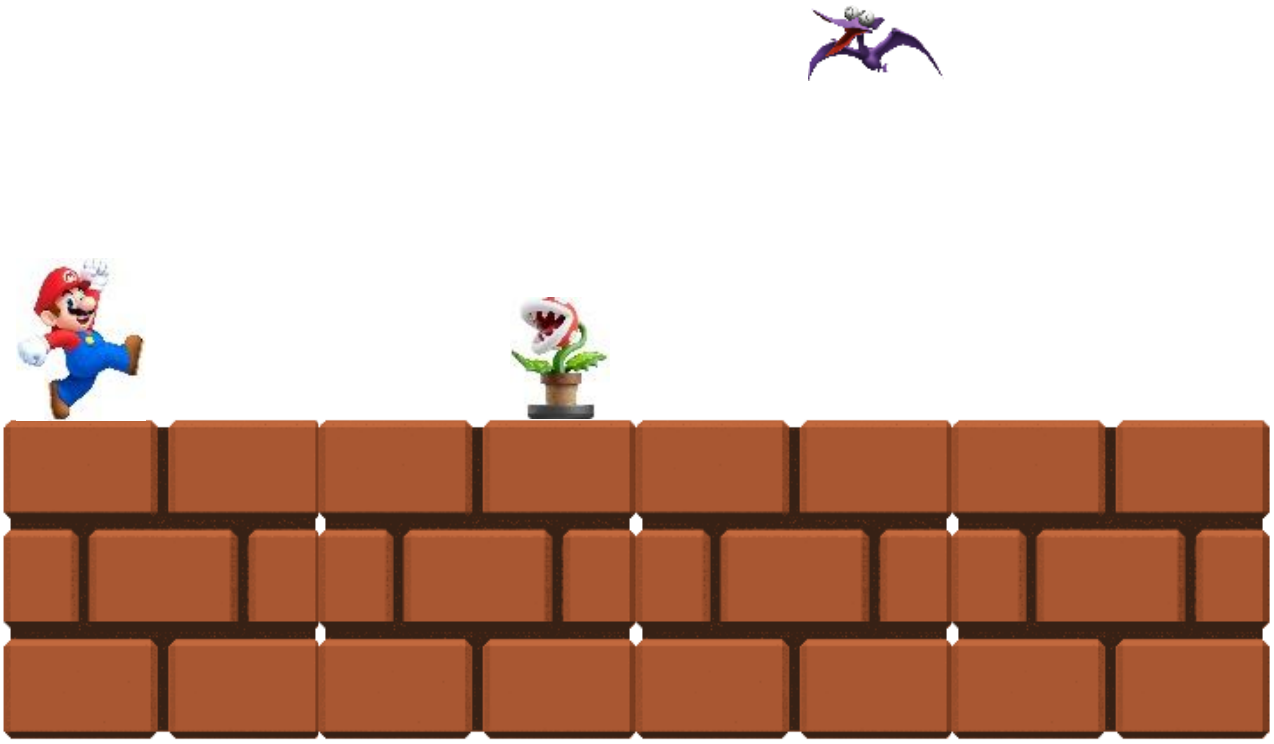
20e édition

18 mars 2026



Le parcours de Super Mario

Le problème : En se déplaçant dans son univers, Super Mario peut rencontrer deux sortes d'obstacles mortels : des fleurs-piranhas sur le sol et des ptérodactyles planant au-dessus de lui. Pour atteindre vivant le bout du chemin, il peut marcher pas-à-pas sous les ptérodactyles sans attirer leur attention ou faire un saut d'une longueur fixe au-dessus des fleurs. Quel sera le nombre minimal de sauts que Super Mario devra effectuer pour arriver vivant au bout de son chemin ?

L'univers de Super Mario est rectiligne et il se déplace toujours dans le même sens (par exemple de gauche à droite). Il ne peut donc jamais reculer. Soit il avance d'un pas, soit il saute d'une longueur toujours équivalente à *LG* pas, *LG* étant fourni. S'il rencontre une fleur-piranha en marchant, il meurt. S'il saute et qu'un ptérodactyle se trouve à n'importe quel endroit de son saut, il meurt. Les ptérodactyles planent en l'air sans bouger.

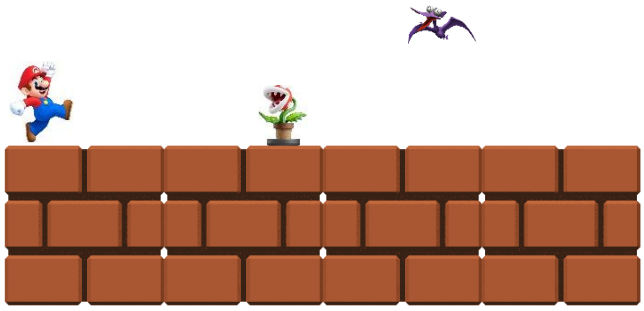
Super Mario démarre toujours du côté gauche, à la position zéro de son univers rectiligne, et pour en sortir doit atteindre le côté droit (la dernière position).

Il n'y a jamais de fleur ou de ptérodactyle sur les premières et dernières cases de l'univers de Super Mario.

De plus, il n'y a jamais de fleur et de ptérodactyle à la même position.

Par exemple dans la situation suivante, avec *LG* = 3, si Super Mario avance de 2 pas pour se placer juste devant la fleur, puis fait un saut de longueur 3 au-dessus de la fleur il sera tué par le ptérodactyle. En revanche, au départ, il peut décider de n'avancer que de 1 pas, puis sauter pour atterrir juste derrière la fleur, puis atteindre le bout en marchant pas à pas sous le ptérodactyle.

LG = 3



Fonctionnement de votre programme

En pratique, votre programme lira sur l'entrée standard (au clavier) 6 lignes.

- Une première ligne contiendra un simple entier (> 1) *LG* : la longueur des sauts.
- La deuxième ligne, contiendra, elle aussi, un entier (> 0) *DIM* : la dimension de l'univers de Super Mario.
- La troisième ligne contiendra un entier (≥ 0) *NbFleurs* : le nombre de fleurs carnivores.
- La quatrième ligne contiendra *NbFleurs* entiers : les positions (0 < pos < *DIM* - 1) des fleurs. Si *NbFleurs* = 0, la ligne est vide.
- La cinquième ligne contiendra encore un entier (≥ 0) *NbPteros* : le nombre de ptérodactyles.
- La sixième ligne contiendra *NbPteros* entiers : les positions (0 < pos < *DIM* - 1) des ptérodactyles. Si *NbPteros* = 0, la ligne est vide.

Votre programme écrira simplement sur la sortie standard (à l'écran) un entier : le nombre minimal de saut nécessaires à Super Mario pour atteindre vivant le bout du chemin, ou le nombre -1 s'il ne peut pas y arriver vivant.

Vous avez la garantie que les données du programme sont bien formées et correctes. Vous ne devez donc pas les vérifier. Par exemple vous avez la garantie que les positions des fleurs et ptérodactyles sont valides (0 < pos < DIM - 1). Vous avez aussi la garantie que ces positions ne se chevauchent pas (sont uniques), que le nombre de fleurs et de ptérodactyles n'excède pas le maximum possible …

Des exemples de programmes, dans divers langages de programmation et respectant le même format d'entrées/sorties, vous sont fournis dans le dossier **`DemosIO`**. **Utilisez-les et ne modifiez pas les entrées/sorties**.

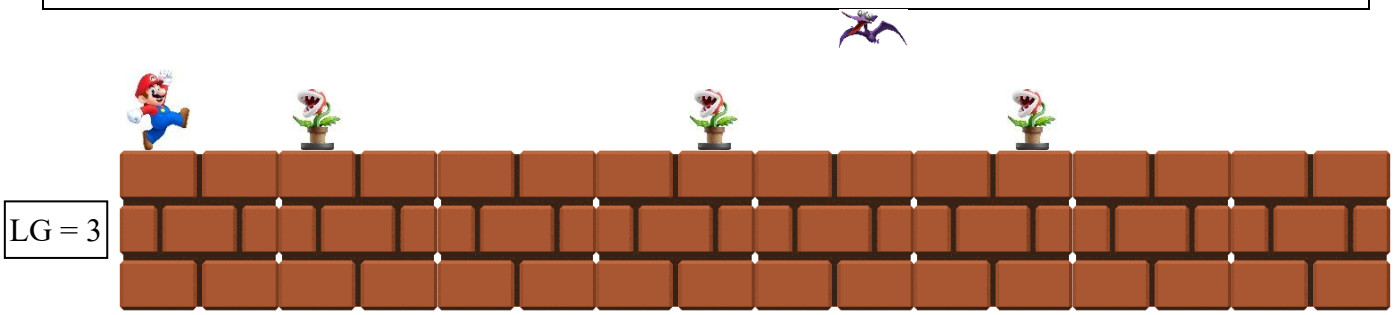
Par exemple :

```
C:\>concours↵
3↵
8↵
1↵
3↵
1↵
5↵
1
```

Dans cet exemple, les saisies de l'utilisateur sont en bleu. Le reste (en gras) est affiché spontanément par l'ordinateur. L'utilisateur entre la longueur de sauts (3) puis la dimension de l'univers (8), le nombre de fleurs (1) et leur positions (ici juste 3) puis le nombre et la (les) position du (des) ptérodactyle(s) (1 ptérodactyle en position 5).

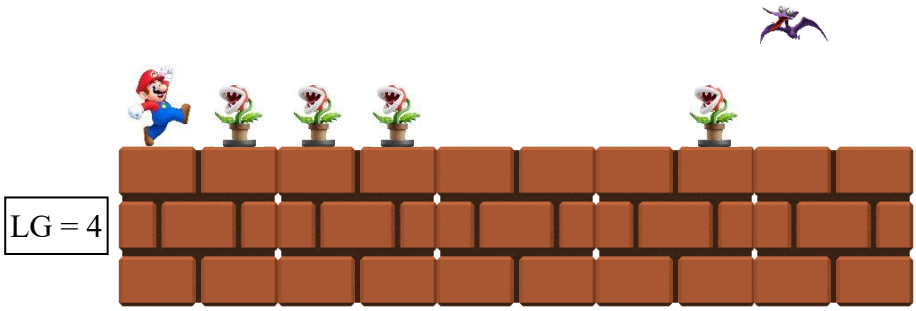
Autres exemples

```
C:\>concours↵
3↵
16↵
3↵
2 7 11↵
1↵
9↵
3
```



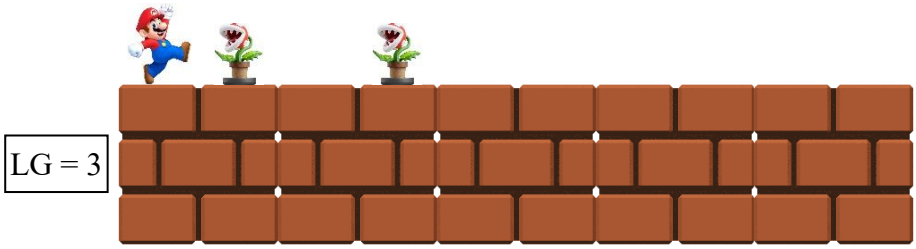
Mario peut immédiatement sauter au-dessus de la 1re fleur pour arriver juste derrière puis avancer de 2 pas (pas 3) puis à nouveau sauter pour arriver juste derrière la 2e fleur, puis avancer de 2 pas sous le ptérodactyle, sauter encore une 3e fois et arriver au bout en marchant. Si après la 1re fleur, Mario avance de 3 pas pour se placer juste devant la 2e fleur et saute au-dessus, il sera tué par le ptérodactyle.

```
C:\>concours↵
4↵
10↵
4↵
1 2 3 7↵
1↵
8↵
-1
```



Mario ne peut pas sauter au-dessus de la dernière fleur tout en évitant le ptérodactyle.

```
C:\>concours↵
3↵
10↵
2↵
1 3↵
0↵
↵
-1
```



Mario ne peut pas bouger.

Afin de tester votre solution, les entrées des exemples ci-dessus, ainsi que quelques autres, sont fournis dans des fichiers textes présents dans le dossier **`DataExemples`**. Le nom du fichier indique aussi la réponse attendue.

Réponses

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

Bon Amusement

Règlement

Date et Lieu : La **20ème édition** du concours de programmation de l'EPFC se déroulera le **mercredi 18 mars 2026 de 13h00 à 21h30** dans les locaux de l'***EPFC*** (Avenue de l'Astronomie, 19 à 1210 Bruxelles – **local B209**). **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) 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 librairies 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 : Une proclamation des résultats avec remise des prix sera organisée **le mercredi 8 avril à 18h**

Elle sera suivie d'un drink offert à tous les participants au concours