

Premiers pas dans la programmation

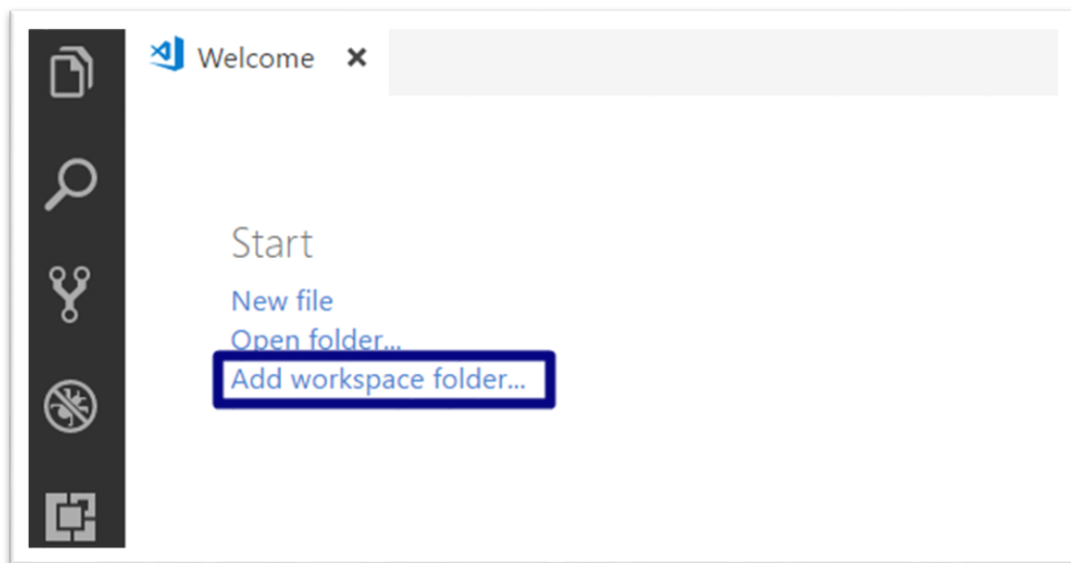
Tâches pour l'exercice en classe et pour les devoirs au cours « Bases de la [programmation](#) » @ SoftUni.

Testez vos solutions dans le **système d'arbitrage** en ligne : <https://judge.softuni.org/Contests/Compete/Index/2399>

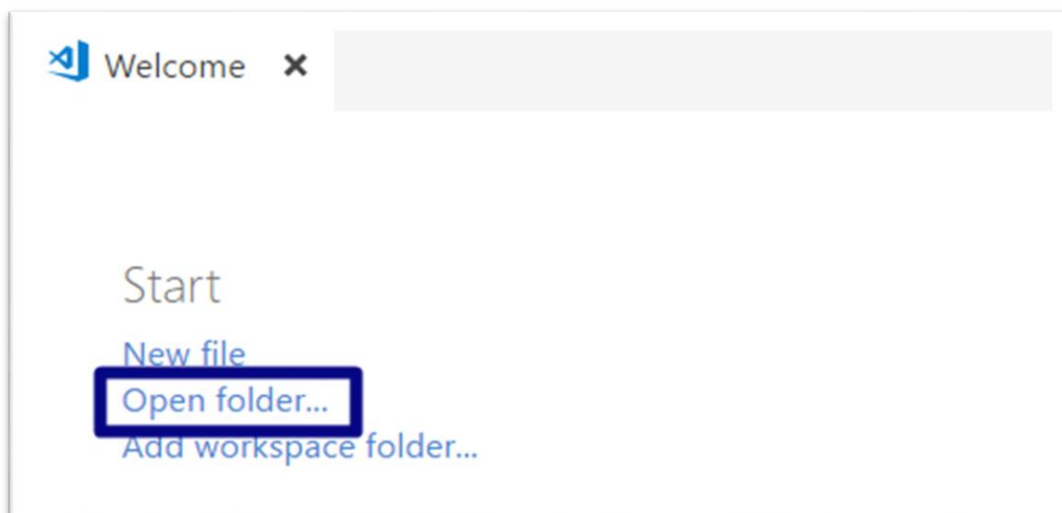
0. Projet Visual Studio Code vierge

Créez un projet vide dans **Visual Studio Code**. Nous combinerons les solutions à toutes les tâches sous la forme de fichiers séparés dans ce projet. Cette possibilité est extrêmement pratique lorsque nous voulons travailler sur plusieurs projets et passer rapidement de l'un à l'autre ou que nous voulons fusionner logiquement plusieurs projets interdépendants. Cela nous aide à séparer les solutions aux tâches et à les conserver, pour les utiliser pour d'autres tâches ou révisions.

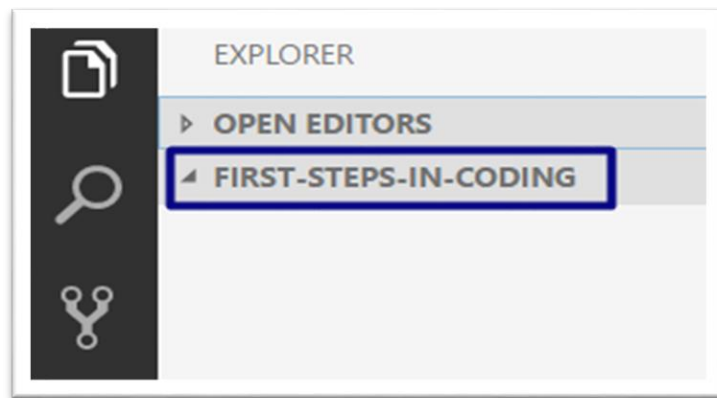
1. Démarrer Visual Studio Code
2. Créez un nouveau dossier qui contiendra les solutions individuelles. Une boîte de dialogue s'ouvrira dans laquelle vous devrez sélectionner son répertoire. Il est recommandé de nommer le dossier en fonction du sujet du travail, par exemple « **Premiers pas dans les calculs** »



Sélectionnez ensuite le dossier en tant qu'environnement de bureau pour ajouter les fichiers de solution JavaScript à vos tâches.



Le panneau de gauche ressemblera à ceci :

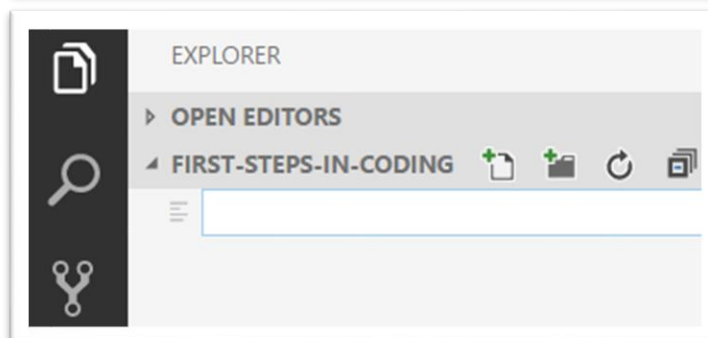
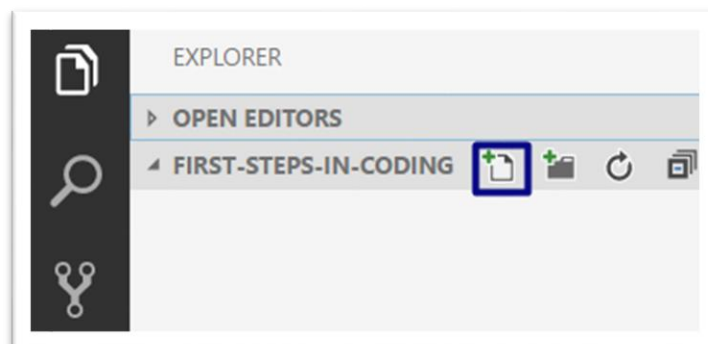


1. Programme console « Hello SoftUni »

Écrivez **une fonction** qui affiche le texte « **Hello SoftUni** ».

Lignes directrices

1. Créez un **nouveau fichier JavaScript** dans le dossier existant et nommez-le de manière appropriée. Il est recommandé que chaque fichier de script soit indiqué comme le nom de la tâche dont il contient la solution.



2. Le contenu du nouveau fichier s'ouvrira dans la fenêtre de **droite**.



3. Allez dans le **fichier hello.js** et créez la **fonction hello()**. Vous pouvez vous aider de l'image ci-dessous :

```
JS hello.js x
1 function hello() {
2   console.log("Hello SoftUni")
3 }
```

4. Lancez le programme avec : **Ctrl + F5**. Pour obtenir un résultat, nous devons « appeler » la fonction :

```
JS hello.js x
1 function hello() {
2   console.log("Hello SoftUni")
3 }
4
5 hello()
```

5. Nous pouvons voir le résultat dans la console ci-dessous :

```
JS hello.js x
1 function hello() {
2   console.log("Hello SoftUni")
3 }
4
5 hello()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Program Files\nodejs\node.exe hello.js
Hello SoftUni

6. Testez la solution à ce problème dans le système d'[arbitrage](#) en ligne :

01. Hello SoftUni

Participants Tests Change Delete

Administration |

```
1 function hello() {
2   console.log("Hello SoftUni")
3 }
```

Allowed working time: 0.100 sec.
Allowed memory: 16.00 MB
Size limit: 16.00 KB
Checker: Trim

JavaScript code (Nod... Submit

Submissions		
<div> ⏮ ⏪ 1 ⏩ ⏭ 🔄 </div>		
Points	Time and memory used	Submission date
✓ 100 / 100	Memory: 11.14 MB Time: 0.140 s	22:51:56 15.02.2018
		<button>Details</button>

2. Nombres de 1 à 10

Écrivez une fonction qui **imprime les nombres de 1 à 10** sur des lignes distinctes de la console.

Lignes directrices

1. Créez un **nouveau fichier JavaScript** dans le dossier existant et nommez-le « **Nums-1-To-10** ». Utilisez le corps de la fonction « **nums1To10()** » :

2. Les commandes de type 10 **console.log()**, l'une après l'autre, pour afficher les nombres de 1 à 10.

```
JS Nums-1-To-10.js
JS Nums-1-To-10.js > nums1to20
1  function nums1to20() {
2      console.log(1);
3      console.log(2);
4      console.log(3);
5      console.log(4);
6      console.log(5);
7      console.log(6);
8      console.log(7);
9      console.log(8);
10     console.log(9);
11     console.log(10);
12 }
```

3. Commencez le programme avec : **Ctrl + F5**. Pour obtenir un résultat, vous devez « appeler » la fonction.

3. Calcul de la face carrée

Écrivez une **fonction** qui **reçoit un entier** et **calcule la face carrée du côté donné**.

Entrée et sortie d'échantillons

entrée	sortie
(["5"])	25

Lignes directrices

1. Acceptez une entrée (un numéro de chaîne) et enregistrez-la dans la variable **a** en la convertissant en un type numérique.
2. Initialiser une variable qui contient la valeur faciale calculée obtenue par la formule **a * a** et affiche le résultat, en se rappelant que pour obtenir un résultat localement, vous devez appeler la fonction et lui transmettre des données d'entrée.

```
function squareArea(input) {  
  let a = Number(input[0]);  
  let area = a * a;  
  console.log(area);  
}
```

4. Convertisseur : pouces en centimètres

Écrivez une fonction qui **lit à partir de la console un nombre réel** et le convertit **en pouces en centimètres**. Pour ce faire, **multipliez les pouces par 2,54** (1 pouce = 2,54 centimètres).

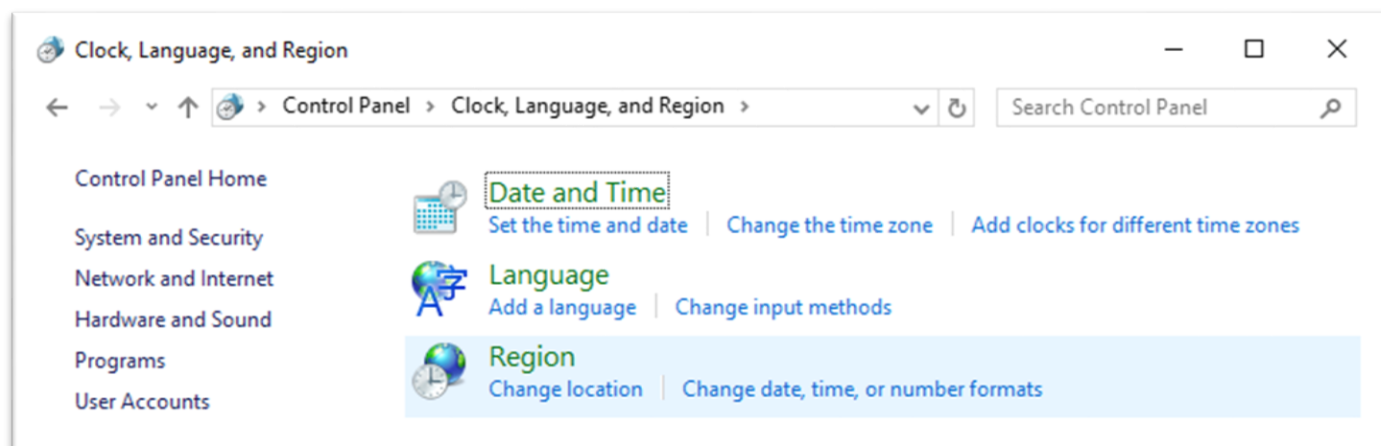
Entrée et sortie d'échantillons

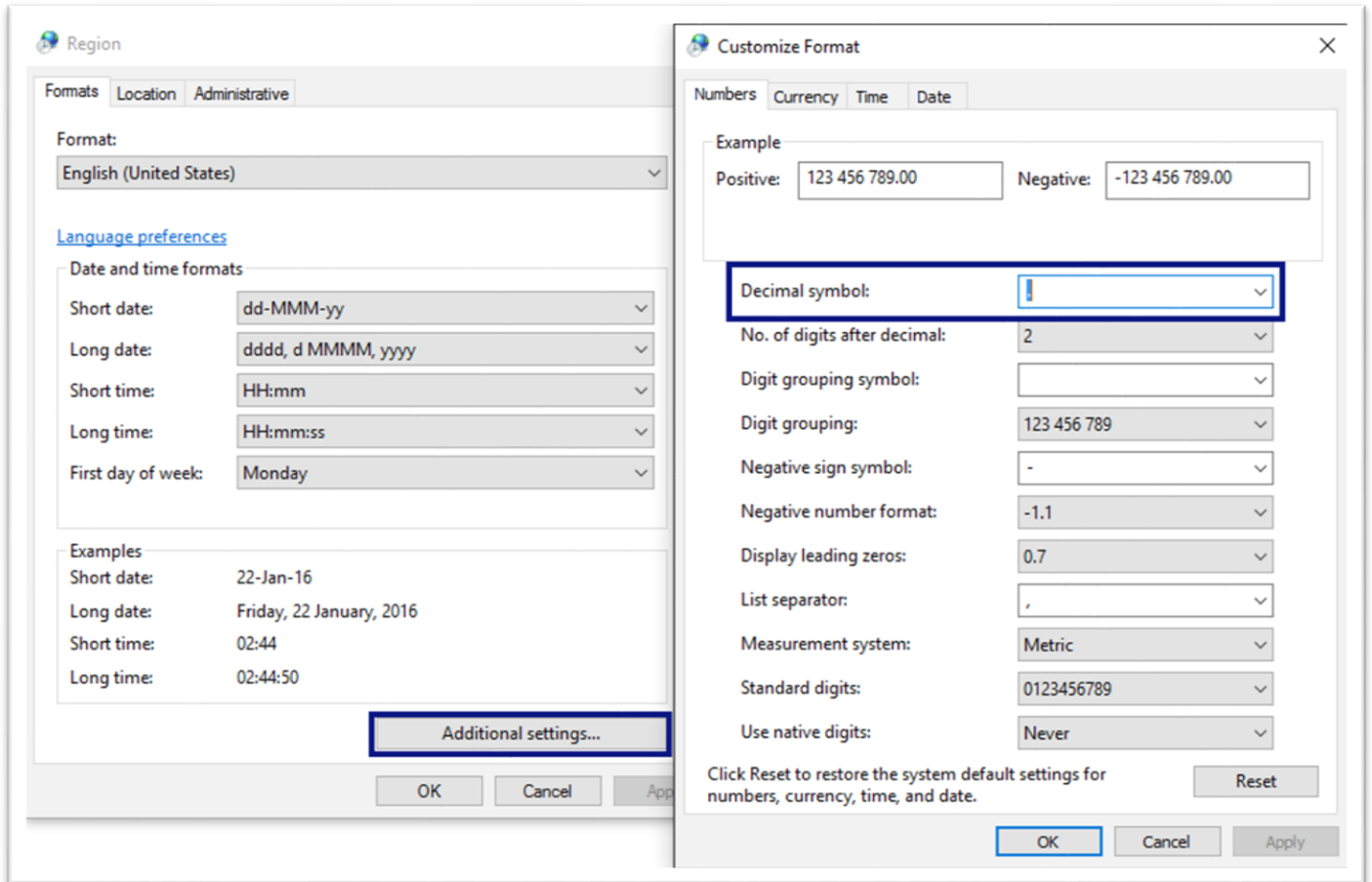
entrée	sortie
["5"]	12.7

entrée	sortie
["7"]	17.78

Attention : en fonction des paramètres régionaux du système d'exploitation, il est possible d'**utiliser** une virgule décimale (paramètres US) au lieu d'une virgule décimale (**paramètres US**). Si le programme attend une virgule décimale et qu'un nombre est entré avec une virgule décimale ou vice versa (une virgule décimale est entrée lorsqu'une virgule décimale est attendue), il ne pourra pas s'exécuter.

Il est recommandé **de modifier les paramètres de votre ordinateur** afin qu'une **virgule décimale** soit utilisée :



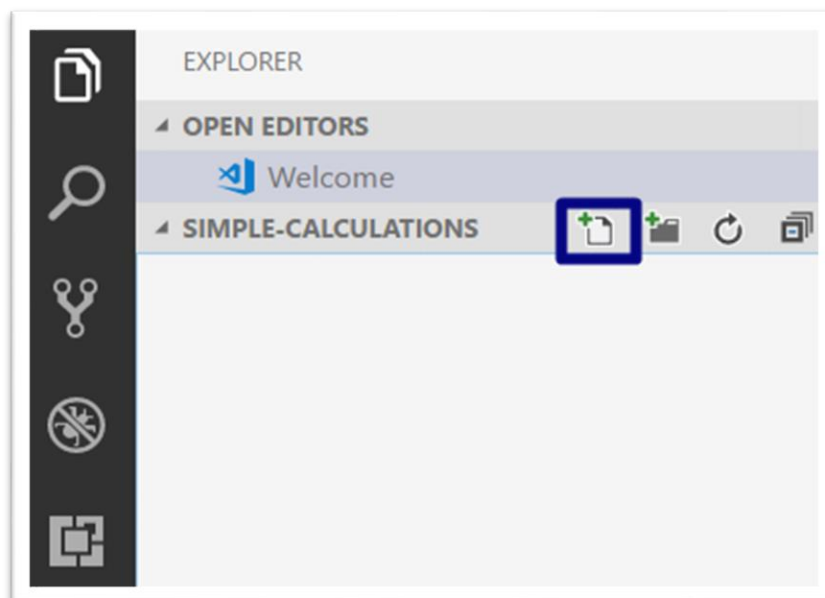


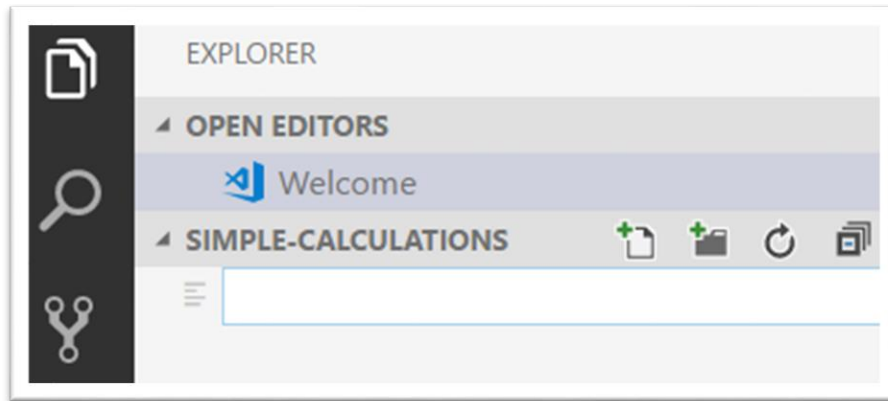
5. Salutations par nom

Écrivez une fonction qui **récupère le nom d'une personne** et affiche « **Bonjour, <nom> !** », où **<nom>** est le nom saisi par la console.

Lignes directrices

1. Tout d'abord, créez **un nouveau fichier JavaScript** dans le dossier existant et nommez-le de manière appropriée. Il est recommandé de dire que chaque fichier de script est le nom de la tâche dont il contient la solution.





2. Le contenu du nouveau fichier s'ouvrira dans la fenêtre de droite



Créez la variable **name** et conservez-y le nom qui est passé par (**input**).

Générez la console à l'aide du modèle suivant :

```
console.log(`Hello, ${name}!`);
```

Comment fonctionne l'exemple ? La méthode **console.log()** nous permet d'écrire entre parenthèses une séquence de caractères qui contient la valeur d'une variable – **\${name}** – et de l'afficher sur la console.

Notez que pour que le texte fonctionne, il doit être entouré du symbole **'**, qui reconnaît le texte brut et une variable. Pour qu'une variable soit reconnue comme telle, elle doit commencer par le **symbole \$** et être placée entre accolades : **{}**.

3. Exécutez le programme avec **Ctrl+F5** et testez avec différents exemples d'entrée.
4. Pour obtenir un résultat, nous devons « appeler » la fonction et lui passer des données d'entrée :

```
function greetings(input) {  
    let name = input[0];  
    console.log(`Hello, ${name}!`);  
}  
greetings(["Niki"]);
```


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\Program Files\nodejs\node.exe Greetings.js  
Hello, Niki!
```

5. Testez votre solution dans judge sans activer l'appel de fonction .

6. Joindre du texte et des chiffres

Écrivez une fonction **qui reçoit : nom, prénom, âge et ville** et affiche un message du type suivant :

« Vous êtes <prénom> <nom>, une personne <âge> ans de <ville>."

Lignes directrices

1. Ajoutez à la solution Visual Studio Code actuelle un autre **fichier JavaScript** nommé « **Concatenate-Data** ».
2. Acceptez l'entrée de la console :

```
function concatenateData(input) {  
  
}
```

3. Affichez la sortie formatée sur la console.
4. Exécutez le programme avec **Ctrl+F5** et testez avec différents exemples d'entrée. Pour générer localement un résultat sur la console, vous devez appeler la fonction et transmettre les données d'entrée dans l'ordre dans lequel vous vous attendez à les recevoir :

```
concatenateData(['Maria', 'Ivanova', 20, 'Sofia'])
```

```
C:\Program Files\nodejs\node.exe Concatenate-Data.js  
You are Maria Ivanova, a 20-years old person from Sofia.
```

7. Préparation des projets

Écrivez un programme qui **calcule le nombre** d'heures qu'il faudra à un architecte pour **préparer les projets** de plusieurs chantiers de construction.

Entrée

À partir de la console, lisez **2 lignes** :

1. **Le nom de l'architecte - texte**
2. **Nombre de projets à préparer - un entier compris entre [0 ... 100]**

Sortie

Sur la console est imprimé :

- « L'architecte aura besoin de {heures requises} heures pour mener à bien {nombre de projets} projet(s). »

Entrée et sortie d'échantillons

Entrée	Sortie	Entrée	Sortie
["Georges » , "4 "]	L'architecte George aura besoin de 12 heures pour réaliser 4 projet(s).	["Sanya » , "9 "]	L'architecte Sanya aura besoin de 27 heures pour réaliser 9 projets.

8. Animalerie

Écrivez un programme qui **calcule le coût d'achat de la** nourriture pour chiens et chats. La nourriture est achetée dans une animalerie, car un paquet de nourriture pour chiens **coûte 2,50 leva** et un paquet de nourriture pour chats **coûte 4 leva**.

Entrée

À partir de la console, lisez **2 lignes** :

1. Le nombre de paquets de nourriture pour chiens – un entier compris dans l'intervalle [0... 100]
2. Le nombre d'emballages de nourriture pour chats – un entier compris entre [0... 100]

Sortie

Sur la console est imprimé :

« {montant final} NV. »

Entrée et sortie d'échantillons

Entrée	Sortie	Entrée	Sortie
["5 , "4 "]	28,5 niv.	["13", "9"]	68,5 niv.

9. Aménagement paysager des cours

Bozhidara possède **plusieurs maisons** sur la côte de la mer Noire et **souhaite aménager les cours de certaines d'entre elles**, créant ainsi **une atmosphère chaleureuse et confortable pour ses invités**.

Écrivez un programme qui **calcule le montant requis** que Bozhidara devra payer à l'entrepreneur. Le prix d'un **mètre carré est de 7,61 BGN avec TVA**. Parce que son chantier est **assez grand**, l'entreprise entrepreneure offre **une réduction de 18% sur le prix final**.

Entrée

Une seule ligne est lue depuis la console :

1. Carré. mètres à aménager – un nombre réel de l'ordre de [0,00 ... 10000.00]

Sortie

Deux lignes sont imprimées sur la console :

- « Le prix final est de : {prix final de la prestation} lv. »
- « La réduction est : {отстъпка} lv. »

Entrée et sortie d'échantillons

Entrée	Sortie	Explications
["550"]	Le prix final est de : 3432.11 lv. La réduction est de : 753,39 lv.	Calculez le coût de l'aménagement paysager de l'ensemble de la cour : $550 * 7,61 = 4185,50$ лв. Nous déduisons la réduction (18 % = 0,18) du total : $0,18 * 4185,5 = 753,39$ лв. Nous calculons le prix final de la prestation : $4185,50$ à $753,39 \rightarrow 3432,11$
Entrée	Sortie	
["150"]	Le prix final est de : 936,03 lv. La réduction est de : 205,47 lv.	Calculez le coût de l'aménagement paysager de l'ensemble de la cour : $150 * 7,61 = 1141,50$ BGN Nous déduisons la réduction (18 % = 0,18) du total : $0,18 * 1141,50 = 205,47$ BGN Nous calculons le prix final de la prestation : $1141,50$ à $205,47 \rightarrow 936,03$