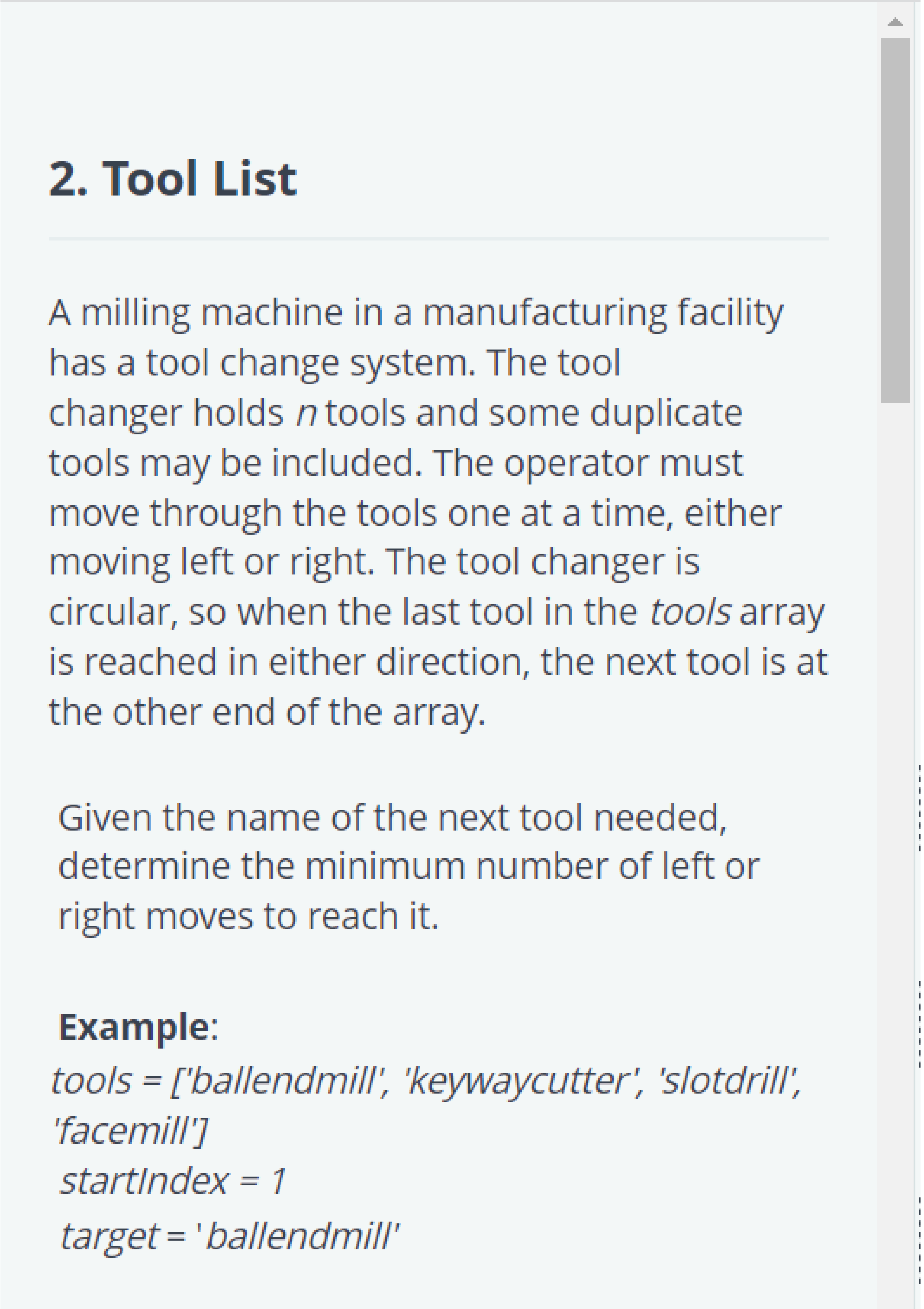


2. Tool List

A milling machine in a manufacturing facility has a tool change system. The tool changer holds *n* tools and some duplicate tools may be included. The operator must move through the tools one at a time, either moving left or right. The tool changer is circular, so when the last tool in the *tools* array is reached in either direction, the next tool is at the other end of the array.

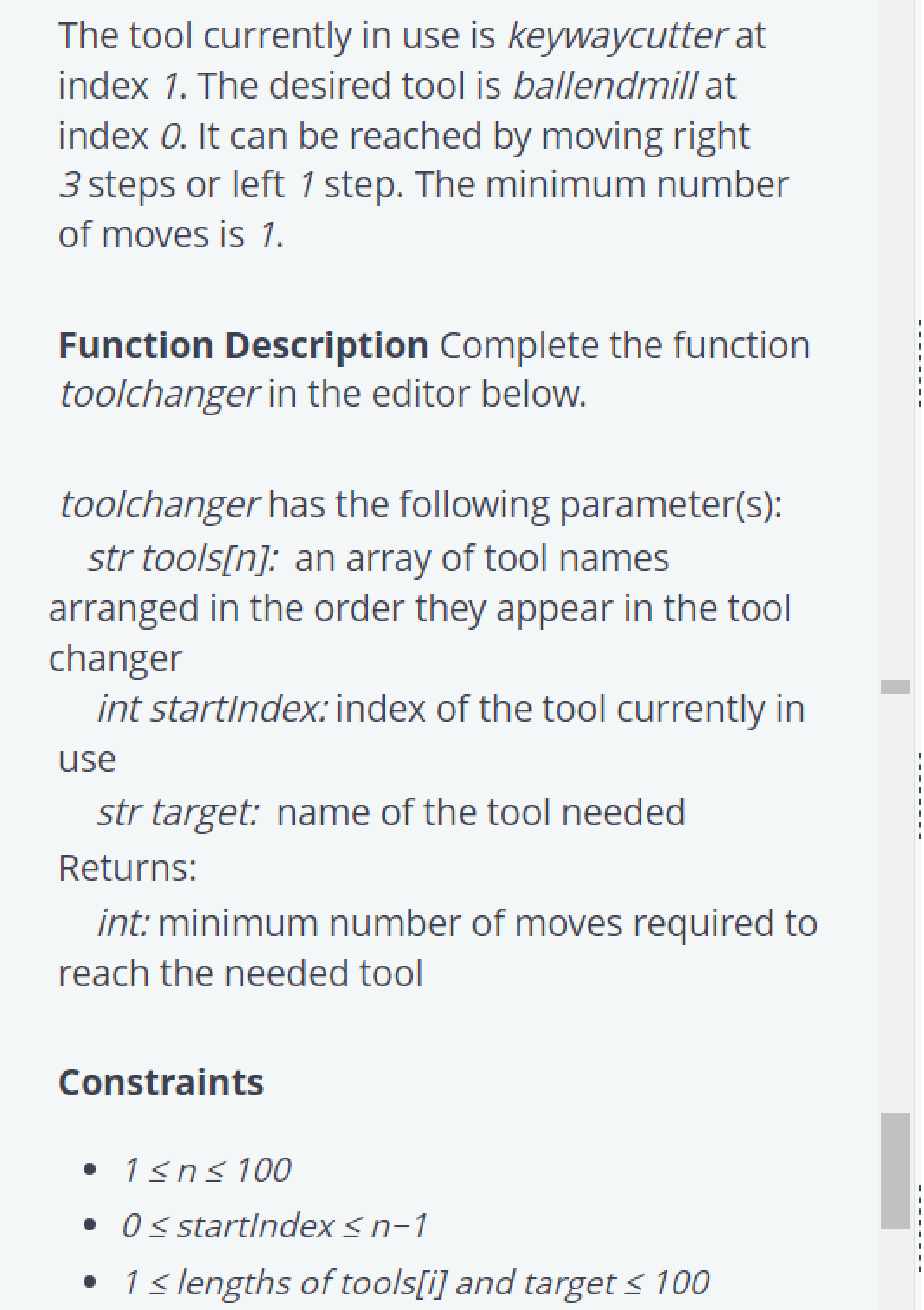
Given the name of the next tool needed, determine the minimum number of left or right moves to reach it.

Example:

tools = ['ballendmill', 'keywaycutter', 'slotdrill', 'facemill']

startIndex = 1

target = 'ballendmill'



The tool currently in use is *keywaycutter* at index *1*. The desired tool is *ballendmill* at index *0*. It can be reached by moving right *3* steps or left *1* step. The minimum number of moves is *1*.

Function Description Complete the function *toolchanger* in the editor below.

toolchanger has the following parameter(s):

str tools[n]: an array of tool names arranged in the order they appear in the tool changer

int startIndex: index of the tool currently in use

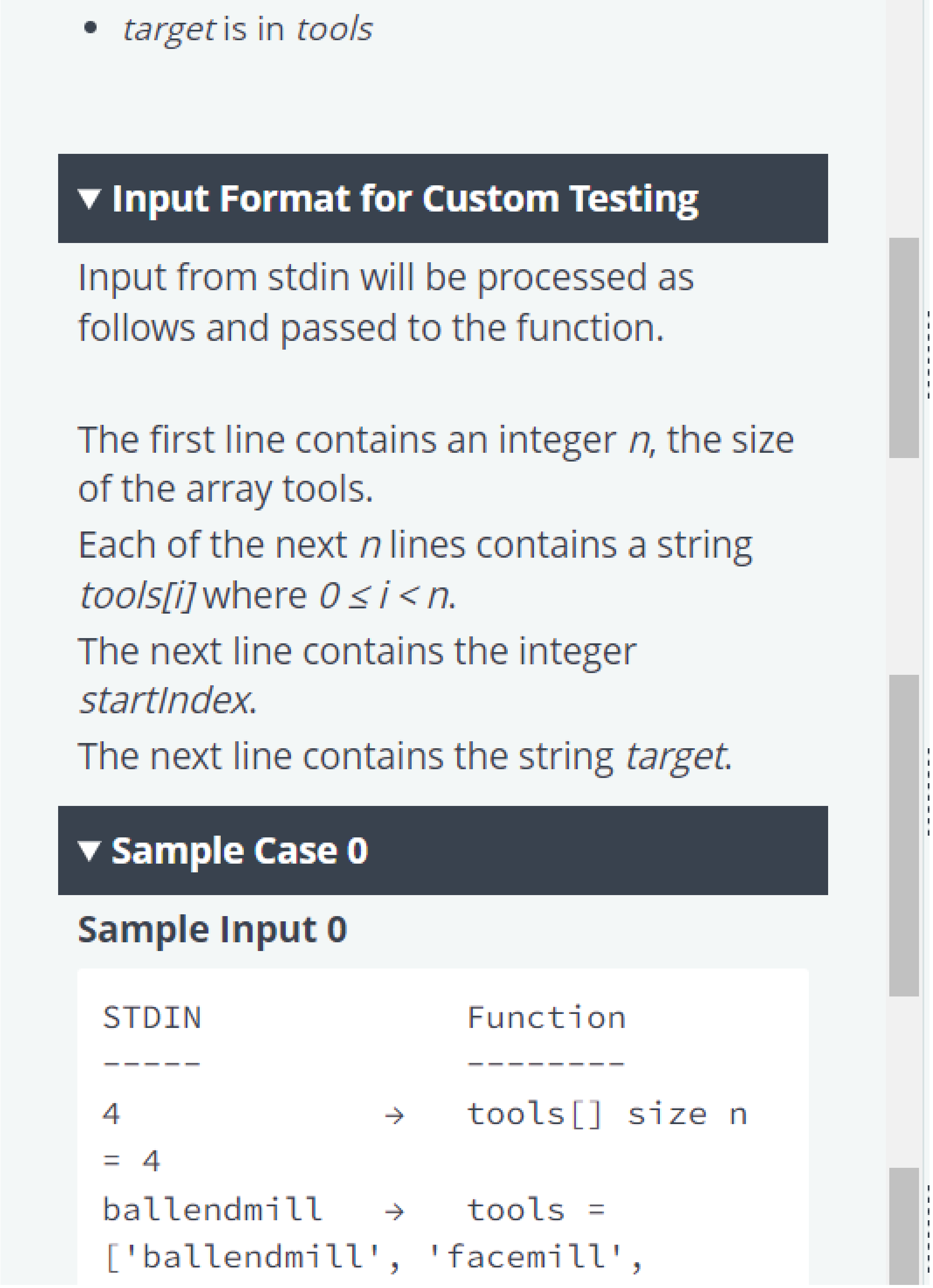
str target: name of the tool needed

Returns:

int: minimum number of moves required to reach the needed tool

Constraints

- $1 \le n \le 100$
- $0 \le startIndex \le n-1$
- $1 \le$ *lengths of tools[i] and target* ≤ 100



- *target* is in *tools*

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array tools.

Each of the next *n* lines contains a string *tools[i]* where *0 ≤ i < n*.

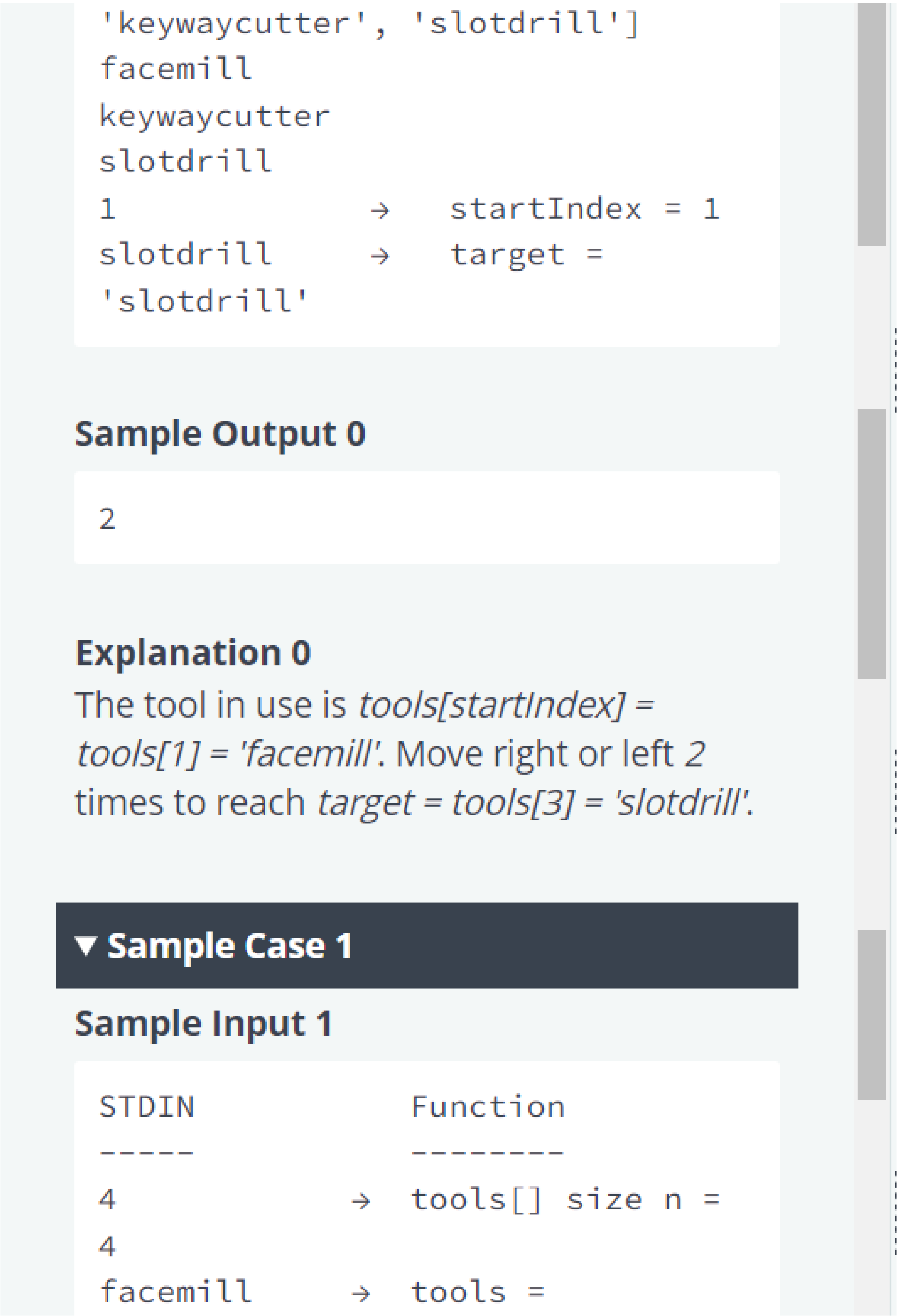
The next line contains the integer *startIndex*.

The next line contains the string *target*.

▼ Sample Case 0

Sample Input 0

```
STDIN                  Function
-----                  --------
4               →      tools[] size n
= 4
ballendmill     →      tools =
['ballendmill', 'facemill',
```



```
'keywaycutter', 'slotdrill']
facemill
keywaycutter
slotdrill
1               →   startIndex = 1
slotdrill       →   target =
'slotdrill'
```

Sample Output 0

```
2
```

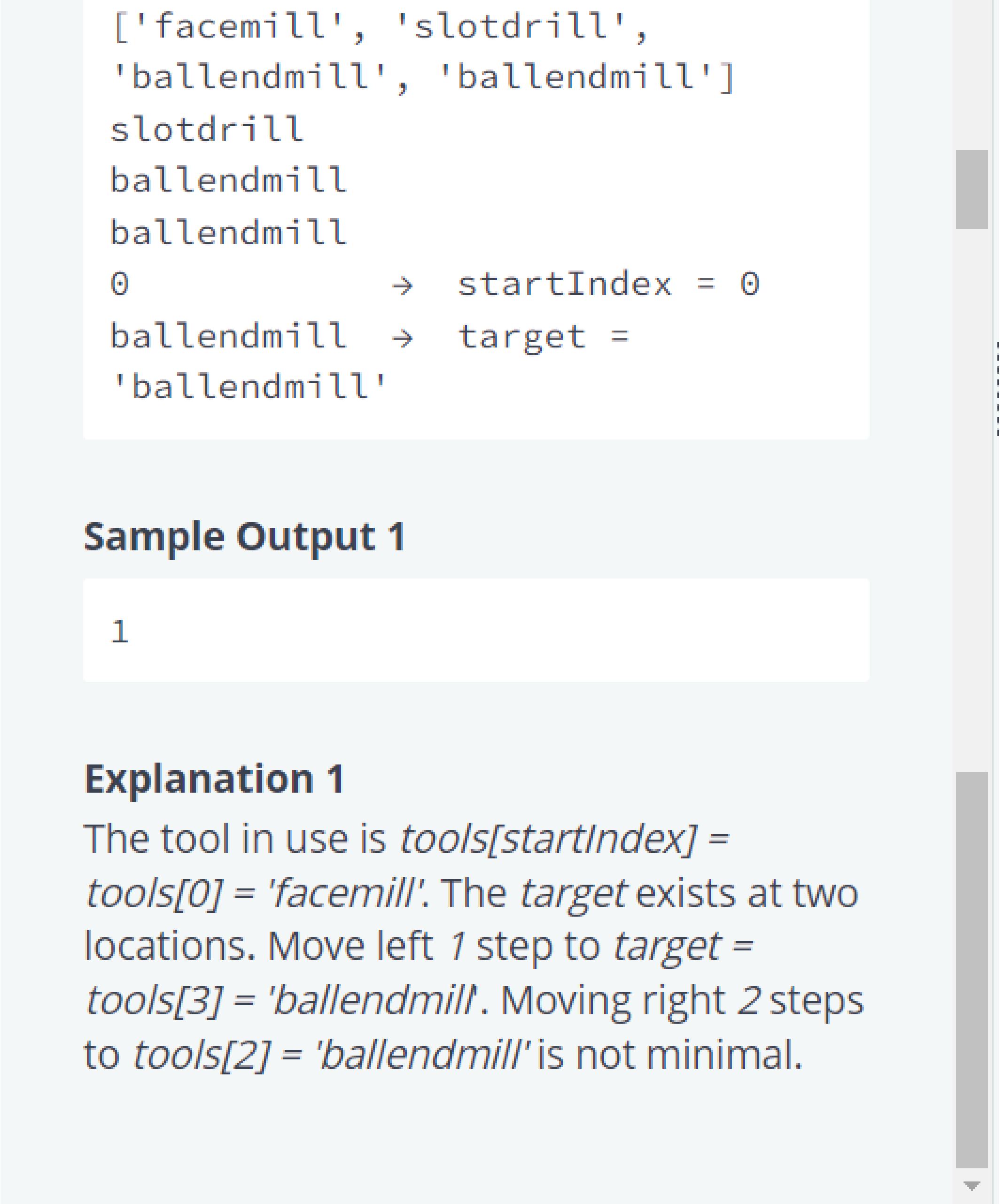
Explanation 0

The tool in use is *tools[startIndex] = tools[1] = 'facemill'*. Move right or left *2* times to reach *target = tools[3] = 'slotdrill'*.

▼ Sample Case 1

Sample Input 1

```
STDIN           Function
-----           --------
4             → tools[] size n =
4
facemill      → tools =
```



```
['facemill', 'slotdrill',
'ballendmill', 'ballendmill']
slotdrill
ballendmill
ballendmill
0            →  startIndex = 0
ballendmill  →  target =
'ballendmill'
```

Sample Output 1

```
1
```

Explanation 1

The tool in use is *tools[startIndex] = tools[0] = 'facemill'*. The *target* exists at two locations. Move left *1* step to *target = tools[3] = 'ballendmill'*. Moving right *2* steps to *tools[2] = 'ballendmill'* is not minimal.