

Intro to the HackerRank Platform



What we'll cover here

1. The HackerRank Welcome Page
2. HackerRank Survey Table of Contents
3. Coding Questions Overview
4. Coding Editor
5. Tips for Working on the Technical Skills Survey

The HackerRank Welcome Page



Sample Test

3 QUESTIONS • 30 MINUTES • PLATFORM HELP

Instructions

1. This is a sample test to help you get familiar with the HackerRank test environment.
2. The test has two coding questions and one multiple choice question. Actual tests may have different number and type of questions.
3. One coding question provides input/output code stubs, and the other does not. You may encounter either situation in a real test.
4. If the code area contains the function signature, just complete the function alone, we'll take care of the main function, headers, etc.
5. If you're expected to write the full code, it'd be mentioned in the code area. All inputs are from STDIN and output to STDOUT. If you're using Java, use the classname as 'Solution.'
6. You can print to console to debug your code using the appropriate print command for each language (Eg: cout for C++, printf for C, etc.)
7. To understand more about the environment, time limits, etc. you can check our [Environment](#) page or read the [FAQ](#).
8. Do not close the browser window/tab before you submit your final answers. If you do so, we cannot guarantee that your work will be saved.
9. [Solve code challenges](#) on HackerRank if you'd like to practice more before taking the real test.

Recommended tips

1. Please close all other applications and tabs before taking the test.
2. We strongly recommend you to take the test in an incognito/private tab to avoid any of your extensions causing issues while taking your tests.

Create a HackerRank account to take Springboard's code challenge

[Sign up](#)[Log in](#)

By signing up you agree to our [Terms of Service](#) and [Privacy Policy](#)

- **Instructions Section** — Each test housed on HackerRank has an introduction that lists the instructions you will need to follow to complete the Skills Survey. Be sure to carefully read these instructions.
- **Signing Up for HackerRank** — Once you have read the instructions, sign up for HackerRank by filling out the text boxes at the bottom of the page and clicking **Sign Up**.



You will have 3 hours to complete the Skills Survey, though we estimate that it will likely take you 1 to 2 hours.

HackerRank Survey Table of Contents

The table of contents is where you will see a list of the questions to complete.

Check the top of the screen to see how much time you have left (you will have 3 hours)

This is where you can track the number of questions you've worked on

To start working on a question, click '**Solve Question**'. Once you have worked on a question, this button will say '**Modify Submission**'. Use this to change an answer. You can modify your answer as much as you like until you click 'I am done with the test'

The screenshot shows the HackerRank interface for a 'Sample Test'. At the top, a dark header bar contains the HackerRank logo, the test name 'Sample Test', a timer showing '28m to test end', and a progress indicator '0/3 Attempted'. Below the header is a table with columns: 'QUESTION NAME', 'TYPE', 'STATUS', and 'Solve Question'. The table lists three questions: Q1 'Find the number!', Q2 'Odd Numbers!', and Q3 'Which of the following sorting algorithms has the best asymptotic runtime comple..'. Each question has a 'Solve Question' button. At the bottom of the table is a button with a checkmark and the text 'I am done with the test'. Annotations with green arrows point to the timer, the progress indicator, the question name, the 'Solve Question' button, and the 'I am done with the test' button.

QUESTION NAME	TYPE	STATUS	
Q1 Find the number!	Coding	not answered	Solve Question
Q2 Odd Numbers!	Coding	not answered	Solve Question
Q3 Which of the following sorting algorithms has the best asymptotic runtime comple..	Multiple Choice	not answered	Solve Question

I am done with the test


You can also click the name of the question to start working on it



When you are done with the survey, click '**I am done with the test**'. Note that you will not be able to change your answers once you have clicked this button.


Coding Questions Overview

Each coding question includes instructions and an editor (aka a terminal) in which you will complete your code. We will take an in-depth look at the terminal in the next slide. The HackerRank Sample Test also includes an overview of the editor.

Instructions

 Sample Test 11m to test end 0/3 Attempted elamson38@gmail.com






 Find the number!

Given an unsorted array of n elements, find if the element k is present in the array or not.

Complete the `findNumber` function in the editor below. It has 2 parameters:

1. An array of integers, `arr`, denoting the elements in the array.
2. An integer, `k`, denoting the element to be searched in the array.

The function must return a string "YES" or "NO" denoting if the element is present in the array or not.



Input Format

The first line contains an integer n , denoting the number of elements in the array `arr`.
Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer describing `arr`.
The next line contains an integer `k`, the element that needs to be searched.


Constraints






- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 10^9$

Output Format

The function must return a string "YES" or "NO" denoting if the element is present in the array or not. This is printed to stdout by locked stub code in the editor.

Editor (aka Terminal)

 Sample Test 10m to test end 0/3 Attempted elamson38@gmail.com



Draft saved 06:14 pm [View Code Diff](#) C++

```
1 #include <bits/stdc++.h>
9
10 // Complete the findNumber function below.
11 string findNumber(vector<int> arr, int k) {
12
13
14 }
15
16 int main() ...
```

Line: 9 Col: 1

☐ Test against custom input [Run Code](#) [Submit code & Continue](#)



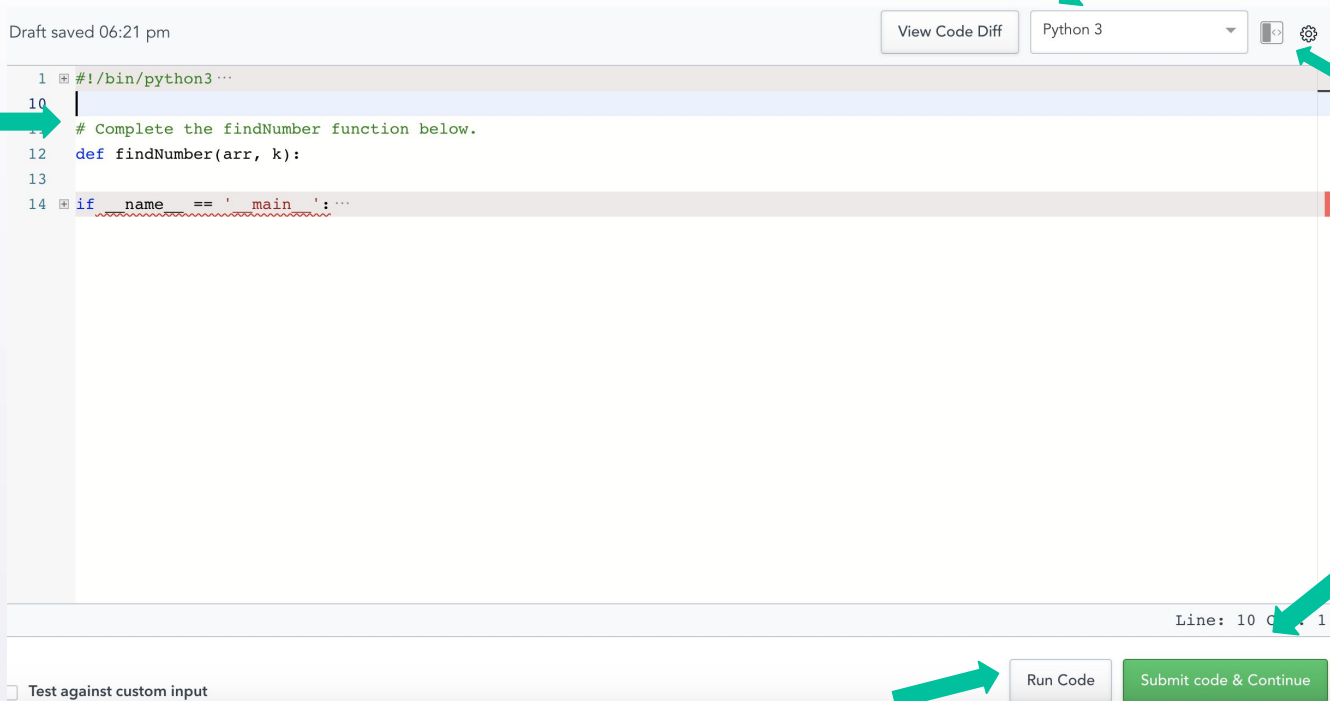
The instructions will often include sample inputs to show you what your answer will likely look like

Coding Editor

The editor is where you will write your code

Code your solutions here. Some questions may have uneditable areas

Use this dropdown menu to select the language you'd like to code in (for you, that will be JavaScript)



Use this button to switch between a vertical and horizontal editor

When you have finalized your code (and tested it by running it), submit your code. You can come back, edit your code, and submit it again as many times as you need. **Be sure to submit your code or your answer won't get saved!**

Use this button to debug code using your own input (this will help you troubleshoot against hidden testcases)

Use the run code button to test your code out. You can run and change your code as many times as you need

STDIN and STDOUT

3h 50m left

ALL

?

4

5

6

7

8

9

10

11

12

Python 3

Autocomplete Ready

🕒 ? ...

1 # Enter your code here. Read input from STDIN. Print output to STDOUT

2 |

STDIN is a file-like object containing a series of inputs, which will be used to test your code. See it as a container holding lots of objects that will be passed to your code.

You can get those inputs into your code with the `input()` function, in statements such as:

```
x = input().
```

Then your code will be tested against all the items in STDIN. For example, suppose your code is just:

```
x = input()
print(x, 'is Great!')
```

And STDIN contains just the strings: 'Tim' and 'Davina'. Then the first test case will output 'Tim is Great!' and the second will output 'Davina is Great!' If this is what the code is supposed to do, then awesome - your code has done the job!

Line: 2 Col: 1

Test Results Custom Input Run Submit Code

This is simply what your program outputs to the console.

Tips for Working on the Technical Skills Survey

1

Carefully read through the instructions to understand the details of the survey, like how much time you have to complete (3 hours) and what to do when you're ready to submit your work.

2

As you work on a question, click the 'Run Code' button to force the code you've written in the code file to run.

- If you run some code and get an error message that you don't understand, Google it!

3

Double check all of your work before you submit your entire survey. However, if you realize later that you gave an incorrect answer, keep in mind that you can retake the survey once you've received your results and have talked to your mentor.