



All Tracks > Basic Programming > Input/Output > Basics of Input/Output



Basic Programming

! Solve any problem to achieve a rank

[View Leaderboard](#)

Topics: Basics of Input/Output

Basics of Input/Output

Tutorial Problems

The very first step of getting started with online judge is to understand:

- How to read input data?
- How to output the answer?

At HackerEarth, input data is read from [standard input](#) stream (STDIN) and results are printed to [standard output](#) stream (STDOUT). Most of the questions will deal with either integers or strings.

An example C code to read an integer from STDIN and printing it out to STDOUT is shown below.

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n); //read input integer from STDIN
    printf("%d",n); //print output integer to STDOUT
    return 0;
}
```

Sample code snippet to read integer for all other languages are given in the code editor below.

An example C code to read a string from STDIN and printing it out to STDOUT.

```
#include <stdio.h>
int main()
{
    char a[100]; //assuming that string size will not exceed 100
    scanf("%s",a); //read a string in array a
    printf("%s",a); //print out array a
}
```

?

```
return 0;  
}
```

Careful not to print any thing else in the output, such as `printf("The answer is %d",n);` because the online judge directly compares your output to expected output like normal string comparison. Say if expected output is `5`, and the code prints `The answer is 5` then when online judge compares `5` with `The answer is 5`, it will say that the code produced wrong result. Every HackerEarth problem has a predefined input and output format.

Every problem will also have **Constraints** section which helps in determining what size of array to be created or what datatypes to use, say `int` or `long long`.

Runtime Errors

While solving the problems on an online Judge, many runtime errors can be faced, which are not clear by the message which comes with them. Lets try to understand these errors.

To get clear about the definition of run time error:

A runtime error means that the program was compiled successfully, but it exited with a runtime error or crashed. You will receive an additional error message, which is most commonly one of the following:

1) SIGSEGV

This is the most common error, i.e., a "segmentation fault". This may be caused e.g. by an out-of-scope array index causing a buffer overflow, an incorrectly initialized pointer, etc. This signal is generated when a program tries to read or write outside the memory that is allocated for it, or to write memory that can only be read. For example, you're accessing `a[-1]` in a language which does not support negative indices for an array.

2) SIGXFSZ

"output limit exceeded". Your program has printed too much data to output.

3) SIGFPE

"floating point error". This usually occurs when you're trying to divide a number by 0, or trying to take the square root of a negative number.

4) SIGABRT

These are raised by the program itself. This happens when the judge aborts your program in the middle of execution. Due to insufficient memory, this can be raised.

5) NZEC

(non-zero exit code) - this message means that the program exited returning a value different from 0 to the shell. For languages such as C/C++, this probably means you forgot to add `"return 0"` at the end of the program. It could happen if your program threw an exception which was not caught. Trying to allocate too much memory during code execution may also be one of the reasons.

For interpreted languages like Python, **NZEC** will usually mean that your program either crashed or raised an uncaught exception. Some of the reasons being in such cases would be: the above mentioned runtime errors. Or, for instance usage of an external library which is causing some error, or not being used by the judge.

6) MLE (Memory Limit Exceeded)

This error means that your program tried to allocate memory beyond the memory limit indicated. This can occur if you declare a very large array, or if a data structure in your program becomes too large.

7) OTHER

This type of error is sometimes generated if you use too much memory. Check for arrays that are too large, or ?

other elements that could grow to a size too large to fit in memory. It can also be sometimes be generated for similar reasons to the **SIGSEGV** error.

Some ways to avoid runtime errors:

- 1) Make sure you aren't using variables that haven't been initialized. These may be set to 0 on your computer, but aren't guaranteed to be on the judge.
- 2) Check every single occurrence of accessing an array element and see if it could possibly be out of bounds.
- 3) Make sure you aren't declaring too much memory. 64 MB is guaranteed, but having an array of size $[100000] * [100000]$ will never work.
- 4) Make sure you aren't declaring too much stack memory. Any large arrays should be declared globally, outside of any functions, as putting an array of 100000 ints inside a function probably won't work.

The below exercise would be a great start. Welcome to the world of Competitive Programming.

Contributed by: Ravi Ojha

Did you find this tutorial helpful?



TEST YOUR UNDERSTANDING

Read from STDIN, Write to STDOUT

Read different types of data from standard input, process them as shown in output format and print the answer to standard output.

Input format:

First line contains integer N .

Second line contains string S .

Output format:

First line should contain $N \times 2$.

Second line should contain the same string S .

Constraints:

$$0 \leq N \leq 10$$

$$1 \leq |S| \leq 15 \text{ where } S \text{ length of string } S$$

SAMPLE INPUT



```
5
helloworld
```

SAMPLE OUTPUT



```
10  
helloworld
```

Enter your code or [Upload your code](#) as file.[Save](#)

Bash (GNU bash, version 4.3.0) ▾



```
1 # Sample bash code  
2
```

1:1 vscode

COMPILE & TEST

SUBMIT

[View all comments](#)

For Developers

Practice programming
Complete reference to
competitive programming
Competitive coding
challenges
Code Monk
Start a programming club

+1-650-461-4192

contact@hackerearth.com

Developer Resources For Business Company

Developers blog	Assess developers	About us
Learn to code by competitive programming	Conduct remote interviews	Press
Developers wiki	Assess university talent	Careers
How to conduct a hackathon		Contact us
		Technical support ?



Organize
hackathon