**Lab: While Loop**

Problems for in-class and homework exercises for the course ["Programming Basics" @ SoftUni](https://softuni.org/).

**Test** your solutions in the **Judge** system: <https://judge.softuni.org/Contests/3495/While-Loop-Lab-PS>

## Read Text

Write a function that reads the elements of an array until it receives the command "**Stop**".  
Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| solve(["Smith",  "SoftUni",  "London",  "Bulgaria",  "SomeText",  "Stop",  "AfterStop",  "Europe",  "HelloWorld"]) | Smith  SoftUni  London  Bulgaria  SomeText |  | solve(["London",  "Berlin",  "Moscow", "Athens",  "Madrid",  "London",  "Paris",  "Stop",  "AfterStop"]) | London  Berlin  Moscow Athens  Madrid  London  Paris |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#0>

## Password

Write a function that initially reads the username and password of a user profile. Then reads the login password.

* On entering the **wrong** passwords: prompt the user to enter a new password.
* On entering the **correct** password: print "Welcome {username}!".

### Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| solve(["Smith",  "1234",  "Pass",  "1324",  "1234"]) | Welcome Smith! | solve(["George",  "secret",  "secret"]) | Welcome George! |

### Hints and Guidelines

1. Initialize two variables **username** and **password**, which will contain the username and password:

Screenshot_1.png

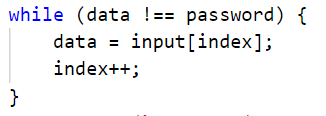
1. Initialize the variable **data**, which will hold the user's login password:

Screenshot_2.png

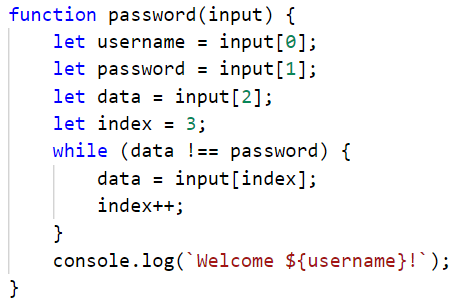
1. Initialize the variable **index**, which will hold the index of the current element in the array. Since we have already assigned the first three elements, we will set **index** to equal **3**.

Screenshot_12.png

1. In a **while** loop, until a valid password is entered, read a new one and increment **index** by 1:



1. When a **valid password** is entered, **print the successful login message:**



### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#1>

## Sum Numbers

Write a function that reads an integer from an array and then integers until their sum becomes greater than or equal to the initial number. When finished, print **the sum of the entered numbers**.

Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| solve([100,  10,  20,  30,  40]) | 100 |  | solve([20,  1,  2,  3,  4,  5,  6]) | 21 |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#2>

## Sequence 2k + 1

Write a function solve(n) that accepts the number **n** entered by the user and prints **all numbers ≤ n in the sequence**: 1, 3, 7, 15, 31... Each following number is calculated by multiplying **the previous** number by **2** and adding **1**.

### Sample Input and Output

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| solve(3) | 1  3 | solve(8) | 1  3  7 | solve(17) | 1  3  7  15 | solve(31) | 1  3  7  15  31 |

### Hints and Guidelines

1. **Create a variable that will be a counter and have an initial value of 1.**
2. **Create a while** loop that repeats until the **counter is smaller** than the number you read from the console.
3. At each loop iteration, **print the counter value** and **add the given value** to it.

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#3>

## Account Balance

Write a function that calculates how much total money is in a bank account after making a certain number of deposits. In each argument, you will get the amount you need to deposit into the account **until you receive a "NoMoreMoney" command**. For each amount received, the console should output **"Increase: "** + the amount and that amount should be **added to the account**. If a number **less than 0** is received, the console should output **"Invalid operation!"** and the **program should end**. When the program ends, it should print **"Total: "** + the total amount in the account.

All amounts must be formatted to **two digits after the decimal point**.

### Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| solve([5.51,  69.42,  100,  "NoMoreMoney"]) | Increase: 5.51  Increase: 69.42  Increase: 100.00  Total: 174.93 | solve([120,  45.55,  -150]) | Increase: 120.00  Increase: 45.55  Invalid operation!  Total: 165.55 |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#4>

## Max Number

Write a function that accepts **integers** until it receives the "**Stop**" command and finds **the largest** among them.

### Sample Input and Output

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| solve([100,  99,  80,  70,  "Stop"]) | 100 | solve([-10,  20,  -30,  "Stop"]) | 20 | solve([45,  -20,  7,  99,  "Stop"]) | 99 | solve([999,  "Stop"]) | 999 | solve([-1,  -2,  "Stop"]) | -1 |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#5>

## Min Number

Write a function that accepts **integers** until it receives the "**Stop**" command and finds **the smallest** among them.

### Sample Input and Output

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| solve([100,  99,  80,  70,  "Stop"]) | 70 | solve([-10,  20,  -30,  "Stop"]) | -30 | solve([45,  -20,  7,  99,  "Stop"]) | -20 | solve([999,  "Stop"]) | 999 | solve([-1,-2,  "Stop"]) | -2 |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#6>

## Graduation

Write a function that calculates the student's **average grade** from all of his/her studies. You will receive the **student's name** as the first argument and his annual grades as the following arguments. The student passes to the next grade **if his annual grade is greater than or equal to 4.00**. If he/she failed more than once, he/she will be excluded and the program ends, printing the **student's name** **and the class he/she was excluded from**.

At the successful completion of grade **12** to be printed:

"{**student's name**} graduated. Average grade: {**average grade of all studies**}"

**In the student is excluded from school, to be printed:**

"{**student's name**} has been excluded at {**class he/she was excluded from**} grade"

**The average grade must be formatted to two digits after the decimal point**.

### Sample Input and Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| solve(["George",  5,  5.5,  6,  5.43,  5.5,  6,  5.55,  5,  6,  6,  5.43,  5]) | George graduated. Average grade: 5.53 | solve(["Mary",  5,  6,  5,  6,  5,  6,  6,  2,  3]) | Mary has been excluded at 8 grade |

### Testing in the Judge System

Test the solution to this problem here: <https://judge.softuni.org/Contests/Compete/Index/3495#7>