Lab: While Loop

Problems for exercise and homework for the "**Programming Basics**" course [@ SoftUni Global](https://softuni.org).

Submit your solutions to the **SoftUni Judge** system at: https://judge.softuni.org/Contests/3695

1. Read Text

Write a program that reads text from the console (string) and prints it until it receives the "**Stop**" command.

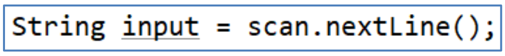
Sample Input and Output

Input	Output
Peter SoftUni Washington Bulgaria SomeText Stop AfterStop Europe HelloWorld	Peter SoftUni Washington Bulgaria SomeText

Input	Output
Washington Berlin Moscow Athens Madrid London Paris Stop AfterStop	Washington Berlin Moscow Athens Madrid London Paris

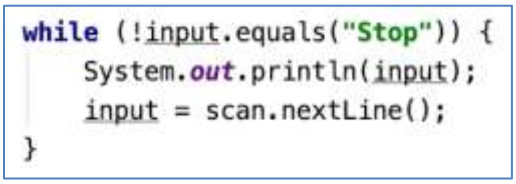
Hints and Guidelines

1. Initialize an **input** variable that will hold the word:



```
String input = scan.nextLine();
```

2. In a **while loop**, until you enter a "**Stop**" command, read a new word and print it to the console:



```
while (!input.equals("Stop")) {
    System.out.println(input);
    input = scan.nextLine();
}
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#0

2. Password

Write a program that reads a **username** and **password**. Then type the password again and check if it is correct

- If the password is **invalid**: the user must type **again** his password.
- If the password is **correct**: print **`"Welcome {username}!"`**.

Sample Input and Output

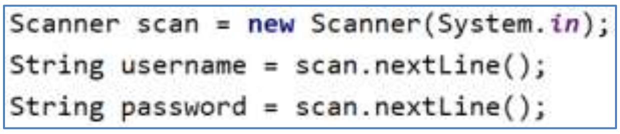
Input	Output

Input	Output

Peter 1234 pass 1324 1234	Welcome Peter!	William secret secret	Welcome William!

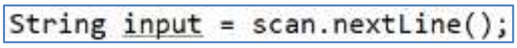
Hints and Guidelines

1. Initialize **username** and **password** variables that will contain the values:



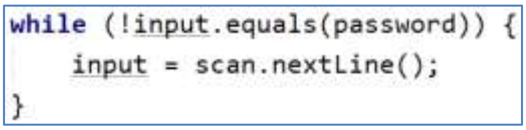
```
Scanner scan = new Scanner(System.in);
String username = scan.nextLine();
String password = scan.nextLine();
```

2. Initialize an **input** variable that will hold the repeated login password:



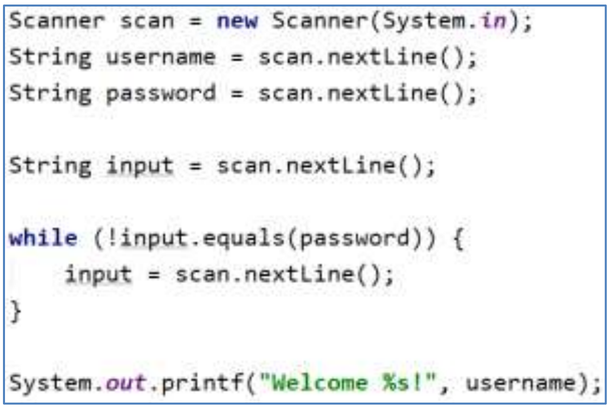
```
String input = scan.nextLine();
```

3. In the **while loop**, until you enter a valid password, read it again:



```
while (!input.equals(password)) {
    input = scan.nextLine();
}
```

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



```
Scanner scan = new Scanner(System.in);
String username = scan.nextLine();
String password = scan.nextLine();

String input = scan.nextLine();

while (!input.equals(password)) {
    input = scan.nextLine();
}

System.out.printf("Welcome %s!", username);
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#1

3. Sum Numbers

Write a program that reads an integer from the console and integers on each line until their sum is greater than or equal to the original number. At the end of the reading, print the **sum of the entered numbers**.

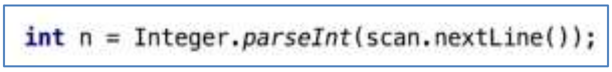
Sample Input and Output

Input	Output	Input	Output

100 10 20 30 40	100	20 1 2 3 4 5 6	21

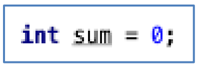
Hints and Guidelines

1. Initialize variable **n** - the number originally read from the console.



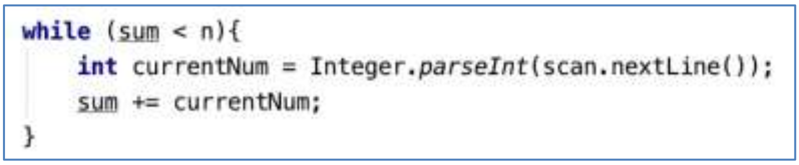
```
int n = Integer.parseInt(scan.nextLine());
```

2. Initialize the variable **sum** in which we will add the read numbers:



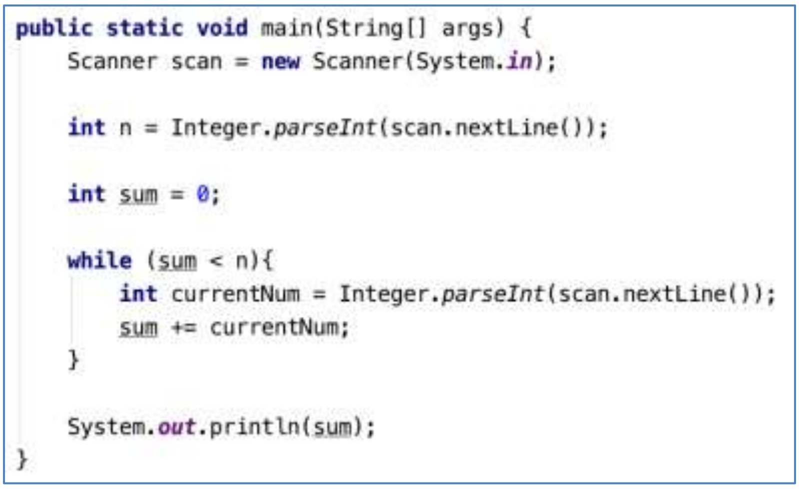
```
int sum = 0;
```

In a **while loop**, until the value of the originally read number is reached, read a new number and add it to the sum::



```
while (sum < n){
    int currentNum = Integer.parseInt(scan.nextLine());
    sum += currentNum;
}
```

3. Print the **sum of the numbers** when it becomes **equal to or greater than** the number originally entered:



```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);

    int n = Integer.parseInt(scan.nextLine());

    int sum = 0;

    while (sum < n){
        int currentNum = Integer.parseInt(scan.nextLine());
        sum += currentNum;
    }

    System.out.println(sum);
}
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#2

4. Sequence 2k+1

Write a program that reads the number **n** entered by the user and prints **all the numbers ≤ n from the sequence: 1, 3, 7, 15, 3…** Each next number is calculated by **multiplying** the **previous one** by **2** and **adding 1**.

Sample Input and Output

Input	Output	Input	Output	Input	Output	Input	Output

3	1 3

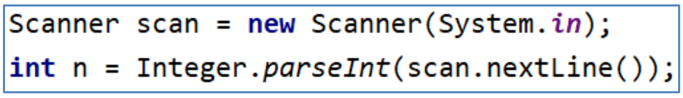
8	1 3 7

17	1 3 7 15

31	1 3 7 15 31

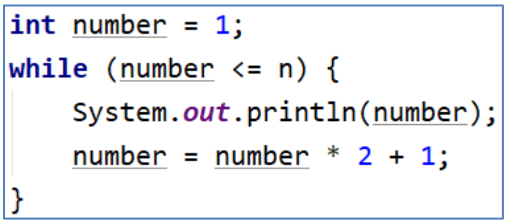
Hints and Guidelines

1. Read **n** - an integer that represents the end of the series



```
Scanner scan = new Scanner(System.in);
int n = Integer.parseInt(scan.nextLine());
```

2. 2. In a **while loop** until **n** is reached, calculate a new number from the sequence and print it:



```
int number = 1;
while (number <= n) {
    System.out.println(number);
    number = number * 2 + 1;
}
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#3

5. Account Balance

Write a program that calculates **how much money** is in the account after you make a **certain number of deposits**. On each line you will receive the amount you need to deposit in the account until you receive the command "**NoMoreMoney**". For each amount received, the console must display "**Increase:** " + the amount and **add it to the account**. If you get a number less than 0, "**Invalid operation!**" must be displayed on the console and the program must end. When the program ends, "**Total:** " must be printed + the total amount in the account **formatted to the second decimal place**.

Sample Input and Output

Input	Output
5.51 69.42 100 NoMoreMoney	Increase: 5.51 Increase: 69.42 Increase: 100.00 Total: 174.93

Input	Output
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/3695#4

6. Max Number

Write a program that, before receiving the "**Stop**" command, reads **integers** entered by the user and **finds the largest** of them. Enter one number per line.

Sample Input and Output

Input	Output
100 99 80 70 Stop	100

Input	Output
-10 20 -30 Stop	20

Input	Output
45 -20 7 99 Stop	99

Input	Output
999 Stop	999

Input	Output
-1 -2 Stop	-1

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#5

7. Min Number

Write a program that, before receiving the "**Stop**" command, reads **integers** entered by the user and finds **the smallest** one of them. Enter one number per line.

Sample Input and Output

Input	Output
100 99 80 70 Stop	70

Input	Output
-10 20 -30 Stop	-30

Input	Output
45 -20 7 99 Stop	-20

Input	Output
999 Stop	999

Input	Output
-1 -2 Stop	-2

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#6

8. Graduation

Write a program that calculates a student's **average grade** throughout his or her **12 years of study**. On the first line you will receive the student's name, and on each line their **annual grades** (12 grades in total). The student goes to the next grade if their annual grade is **greater than or equal to 4.00**. If the student has failed more than once, they is expelled, and the program ends by **printing the student's name** and in which **class he needs to repeat**.

Upon successful completion of 12th grade to print:

"**{name of student} graduated. Average grade: {average grade through the years}**"

In case the student is expelled from school, print:

"**{name of student} has been excluded at {grade he was expelled at} grade**"

The value must be formatted to the second decimal place.

Sample Input and Output

Input	Output
John 5 5.5 6	John graduated. Average grade: 5.53

Input	Output
Sophie 5 6 5	Sophie has been excluded at 8 grade

5.43 5.5 6 5.55 5 6 6 5.43 5	

6 5 6 6 2 3	

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3695#7