

CS101 Midterm Exam

Introduction to Computer Programming

1) String manipulation

Objective: Show that you understand how to correctly use variables and can manipulate strings using Python programming language.

Task: Code the following requirements:

You are given the firstname and lastname of a person on two different lines.

Read these **two** inputs into your program and print the following:

➤ *Hello firstname lastname! You just delved into python.*

Input Format

The first line contains the first name, and the second line contains the last name.

Output Format

Print the output on a single line as noted above.

Sample Input:

Sam
Jones

Sample Output:

Hello Sam Jones! You just delved into python.

2) String manipulation using a function and string functions

Objectives: Show that you understand how to create and use a function, and can manipulate a string using Python built-in string functions.

Task: code the following requirements:

- Create a string of multiple words (i.e.: 'this is a string').
- Write a function which takes the string as a parameter, **splits** the string on the " " (space) delimiter and **joins** the string using a "-*-" . Return the result from your function and print that result.

Sample Input

this is a string

Sample Output

this-*-is-*-a-*-string

3) Iterations and conditional logic

Objective: Show that you understand how to write a loop and a logical condition to achieve a mathematical result.

Task: code the following requirements:

- Read an integer N from the user.
- For all non-negative integers $i < N$, print i raised to the 2nd power.

Sample Input

5

Sample Output

0

1

4

9

16

4) Conditional control flow, functions

Objective: Demonstrate you have an understanding of conditional program flow and control. Apply that knowledge using functions.

Criteria:

We add a Leap Day on February 29, almost every four years. The leap day is an extra, or intercalary day and we add it to the shortest month of the year, February.

In the Gregorian calendar three criteria must be taken into account to identify leap years:

The year can be evenly divided by 4, is a leap year, unless:

The year can be evenly divided by 100, it is NOT a leap year, unless:

The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are **NOT** leap years.

Task: code for the following requirements:

- Get a year from the user
- Write a function to check if the inputted year is a leap year or not based on the above criteria.
- Invoke (call) your function with the test cases provided below and print the results.

Output Format

The function returns (and you print) a Boolean value (True/False)

Sample Input

1990

Sample Output

False

Explanation

1990 is not a multiple of 4 hence it's not a leap year.

Test cases: 2100 → returns **false**

2400 → returns **true**

Note – if you run your function with the test cases and they do not produce the expected results, your algorithm is not correct.

5) Working with Dictionaries, Lists, and Functions

Objective: Demonstrate you understand working with lists, dictionaries and functions.

Task: Write a function which takes a list of strings as a parameter. The function programmatically builds and returns a dictionary where the keys are each of the words from the list parameter and the values are the number of times that word appears in the list. Print the output returned from the function.

Constraints: The list needs to contain at least one repeated word.

Sample Input:

"silly sally sold silly sea shells on the sea shore. sally is so silly"

Sample Output:

```
{'silly': 3, 'sally': 2, 'sold': 1, 'sea': 2, 'shells': 1, 'on': 1, 'the': 1, 'shore.': 1, 'is': 1, 'so': 1}
```

6) Problem Solving

Objective: Demonstrate problem solving using the python tools you've worked with in class.

Criteria:

You have a record (dictionary) of N students. Each record contains the student's name, and their percent marks in Math, Physics and Chemistry. The marks can be floating values.

Task:

The user enters some integer N followed by the names and marks for students. You are required to save the record in a dictionary data type. The user then enters a student's name. Output the average percentage marks obtained by that student, correct to two decimal places

Constraints:

$2 \leq N \leq 10$

$0 \leq \text{Marks} \leq 100$

Input Format

The first line contains the integer N , the number of students. The next N lines contains the name and marks obtained by that student separated by a space. The final line contains the name of a particular student previously listed.

Output Format

Print one line: The average of the marks obtained by the particular student correct to 2 decimal places.

Sample Input

```
3
Sam 67 68 69
Pam 70 98 63
Will 52 56 60
Will
```

Sample Output

```
56.00
```

Explanation

Marks for Will are whose average is 56.00

Sample Input

```
2
Bob 25 26.5 28
Rob 26 28 30
Bob
```

Sample Output

```
26.50
```