Worksheet: Functions

CS 101 Algorithmic Problem Solving

Fall 2023

Name(s): ______________________________

HU ID *(e.g., xy01042)*: ______________________________

1. AC in summer!

Three of your friends are sitting in a room - Ali, Raahim, Saad. They need to decide on the temperature to set on their AC. Everyone has a demand.

- Ali wants the temperature to be at least A degrees.
- Raahim wants the temperature to be at most B degrees.
- Saad wants the temperature to be atleast C degrees.

Write a function temperature which returns a ”Yes” if they can decide on some temperature which fits all their demands or ”NO”, if no temperature fits all their demands.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $1 \leq A, B, C \leq 200$

Interaction

The input comprises a single line containing 3 space-separated integers denoting the values of A, B, C

The output must contain either a ”YES” or ”NO”.

Sample

Input	Output
30 35 25	YES
30 35 40	NO

In the first case, $(A, B, C) = (30, 35, 25)$. Ali wants the temperature to be ≥ 30, Raahim wants it to be ≤ 35 and Saad wants it to be ≥ 25. Temperatures from 30 - 35 satisfy their demands so the answer is a ”YES”.

In the second case, $(A, B, C) = (30, 35, 40)$. Ali awants the temperature to be ≥ 30, Raahim wants it to be ≤ 35 and Saad wants it to be ≥ 40. There is no temperature that satisfies this.

Exercise

In the space provided, indicate the outputs for the given inputs.

Input	Output
30 35 35	YES
30 25 35	NO
19 28 10	YES

Problem Identification

Briefly explain the underlying problem you identified in the above question that led you to your solution.

Input: A, B, C
Output: "Yes" if A is less than or equal to B, and B is greater than or equal to C, "NO" otherwise.

Pseudocode

```
def temperature(A,B,C):
    if A<=B and B>=C:
        return "YES"
    else:
        return "NO"

print(temperature(A,B,C))
```

Dry Run

Using any of the inputs provided in the Exercise section above, dry run your psuedocode in the space below.

With A=30, B=35, and C=35, the code runs as follows:

1. The temperature function is called with A=30, B=35, and C=35.
2. Inside the function, it checks if A (30) is less than or equal to B (35), which is true. It also checks if B (35) is greater than or equal to C (35), which is true as well.
3. Since both conditions are true, the function returns "YES", which is also our expected output.

This means the applied logic is correct

2. Equalizing something

You have two integers A and B. In one operation you can chose any integer d, and make one of the following moves:

- add d to A and subtract d from B.
- Add d to B and subtract d from A.

You can make as many operations as you want.

Write a function equalizer that returns yes if you can make A and B equal and no if you cannot.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $1 \leq A, B \leq 1000$

Interaction

The input comprises a single line containing 2 space-separated integers denoting the values of A and B.

The output must contain either a Yes or a No.

Sample

Input	Output
3 3	Yes
1 2	No

In the first case, $(A, B) = (3, 3)$. A and B are already equal so no operations are needed hence a yes.

In the second case, $(A, B) = (1, 2)$. You cannot make any operations to make A and B equal.

Exercise

In the space provided, indicate the outputs for the given inputs.

Input	Output
1 5	Yes
90 23	No
31 22	No

Problem Identification

Briefly explain the underlying problem you identified in the above question that led you to your solution.

Input: A, B
Output: "Yes" if the absolute difference between A and B is an even number, "No" otherwise.

Pseudocode

```
def equalizer(A,B):
    d = A-B
    if d < 0:
        d = -d
    if (d%2 == 0):
        return "Yes"
    else:
        return "No"

print(equalizer(A,B))
```

Dry Run

Using any of the inputs provided in the Exercise section above, dry run your pseudo code in the space below.

With $A = 1$ and $B = 5$, the code runs as follows:

The `equalizer` function is called with $A = 1$ and $B = 5$.

1. Inside the function, it calculates the absolute difference between A and B: $D = A - B$, which results in $D = 1 - 5$, making $D = -4$.
2. The code then checks if D is less than 0, which is true since D is -4. It proceeds to make d positive by taking its absolute value: $D = -D$, so D becomes 4.
3. Next, it checks if D is even, and in this case, 4 is indeed an even number.
4. Since the condition ($D\%2 == 0$) is true, the function returns "Yes", which is also our expected output.

This means the applied logic is correct

3. Car Mileage matters doesn't it?

You want to go on a road trip and decide to rent a car to travel to a restaurant that is N kilometers away. You can either rent a petrol car or a diesel car.

One liter of petrol would cost you X rupees and one liter of diesel costs Y rupees. You can travel A kilometers with one liter of petrol and B kilometers with one liter of diesel.

You can also buy petrol and diesel in any amount, not necessarily an integer.

Write a function mileage which takes the parameters N, X, Y, A and B and returns diesel if diesel car is better petrol if petrol car is better and any of the costs of both cars are the same.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $1 \leq N, X, Y, A, B \leq 100$

Interaction

The input comprises a single line containing 5 space-separated integers denoting the values of N, X, Y, A, B

The output must contain either a petrol, diesel or any depending on criteria specified above.

Sample

Input	Output
20 10 8 2 4	Diesel
50 12 12 4 3	Petrol

In the first case, $(N, X, Y, A, B) = (20, 10, 8, 2, 4)$. The cost of traveling by the petrol car will be 100 rupees while that of using the diesel car will be 40 rupees. Hence, diesel car is better.

In the second case, $(N, X, Y, A, B) = (50, 12, 12, 4, 3)$. The cost of traveling by the petrol car will be 150 rupees while that of using the diesel car will be 200 rupees. Hence, a petrol car is better.

Exercise

In the space provided, indicate the outputs for the given inputs.

Input	Output
40 3 15 8 40	Any
21 9 9 2 9	Diesel
20 12 20 8 2	Petrol

Problem Identification
Briefly explain the underlying problem you identified in the above question that led you to your solution.

Input: N, X, Y, A, B
Output: 'Diesel' if the relative cost of petrol is higher, 'Petrol' if the relative cost of diesel is higher, 'Any' if both are equal.

Pseudocode

```
def mileage(N,X,Y,A,B):
    reqd_petrol = N/A
    reqd_diesel = N/B
    petrol_cost = reqd_petrol * X
    diesel_cost = reqd_diesel * Y
    if petrol_cost < diesel cost:
        return "Petrol"
    elif petrol_cost == diesel_cost:
        return "Any
    else:
        return "Diesel"

print(mileage(N,X,Y,A,B))
```

Dry Run
Using any of the inputs provided in the Exercise section above, dry run your psuedocode in the space below.

With $N = 40$, $X = 3$, $Y = 15$, $A = 8$, and $B = 40$, the code runs as follows:

1. The `mileage` function is called with $N = 40$, $X = 3$, $Y = 15$, $A = 8$, and $B = 40$.
2. Inside the function, it calculates the required petrol in liters: $reqd\_petrol = \frac{N}{A} = \frac{40}{8} = 5$ liters.
3. It also calculates the required diesel in liters: $reqd\_diesel = \frac{N}{B} = \frac{40}{40} = 1$ liter.
4. The cost of petrol is computed as: $petrol\_cost = reqd\_petrol \cdot X = 5 \cdot 3 = 15$ units.
5. The cost of diesel is calculated as: $diesel\_cost = reqd\_diesel \cdot Y = 1 \cdot 15 = 15$ units.
6. Next, the code compares the petrol cost and diesel cost. Since they are equal (petrol_cost equals diesel_cost), it returns "Any", which is also our expected output for these values.

This means the applied logic is correct

4. Five is a great number.

You are given a positive integer N, you want to determine if its possible to rearrange the digits of N and obtain a multiple of 5.

Write a function five that takes in the parameter N and returns a yes if it is possible to rearrange the digits to obtain a multiple of 5 and no if it is not possible.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $1 \leq N \leq 100000$

Interaction

The input comprises a single integer which is the value of N

The output must contain a either a Yes or No according to the above criteria

Sample

Input	Output
115	Yes
119	No

In the first case, $N = 115$, The number 115 is divisible by 5 and needs no rearranging.

In the second case, $N = 119$, The number cannot be rearranged in any way to make it divisible by 5.

Exercise

In the space provided, indicate the outputs for the given inputs.

Input	Output
103	Yes
158	Yes
291	No

Problem Identification

Briefly explain the underlying problem you identified in the above question that led you to your solution.

Input: N
Output: ”Yes” if N contains ’0’ or ’5’, ”No” otherwise.

Pseudocode

```
def five(N):
    X = str(N)
    flag = 0
    for i in X:
        if (i=="0" or i=="5"):
            flag = 1
    if flag == 0:
        return "No"
    else:
        return "Yes"

print(five(N))
```

Dry Run

Using any of the inputs provided in the Exercise section above, dry run your psuedocode in the space below.

With $N = 103$, the code runs as follows:

1. The `five` function is called with $N = 103$.

2. Inside the function, it converts the integer N into a string X. In this case, X becomes "103."
3. The variable $flag$ is initialized to 0.
4. The code then enters a loop that iterates through each character i in the string X.
5. During each iteration, it checks if i is equal to "0" or "5." If it is, it sets $flag$ to 1.
6. After iterating through all characters in X, if $flag$ remains 0, it returns "No." Otherwise, it returns "Yes."
7. In this specific case, when the loop checks the characters "1," "0," and "3," it finds that the character "0" is indeed equal to "0." so $flag$ is set to 1. As a result, the function returns "Yes", which is also our expected output

This means the applied logic is correct

5. Happy String

You have a string S with you, a string is happy if it contains a contiguous substring of length strictly greater than 2 in which all its characters are vowels.

Write a function happy that takes in the parameter S and returns if the string is happy or not.

Note that, in english alphabetEnglishs are a,e, i,o, and u.

Note: string will only contain lowercase letters.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $3 \leq |S| \leq 1000$

Interaction

The input comprises a string of variable length.

The output must contain either Happy or Sad according to the criteria given above.

Sample

Input	Output
abxy	Sad
abcdeeafg	Happy

In the first case, $S = abxy$, There is only one vowel a therefore the string is sad.

In the second case, $S = abcdeeafg$, There is a 3 vowel substring eea therefore the string is happy.

Exercise

In the space provided, indicate the outputs for the given inputs.

Input	Output
aeiou	Happy
abedfeg	Sad
ewkiaou	Happy

Problem Identification
Briefly explain the underlying problem you identified in the above question that led you to your solution.

Input: S
Output: ”Happy” if three contiguous vowels are encountered in S, ”No” otherwise.

Pseudocode

```
def happy(S):
    count = 0
    flag = False
    for i in S:
        if i in "aeiou":
            count = count + 1
            if count == 3:
                flag = True
                break
        else:
            count = 0
            flag = False

    if flag == True:
        return "Happy"
    else:
        return "Sad"

print(happy(S))
```

Dry Run
Using any of the inputs provided in the Exercise section above, dry run your psuedocode in the space below.

With S = *”abedfeg”*, the code runs as follows:

1. The ‘happy‘ function is called with S = *”abedfeg”*.
2. Inside the function, two variables are initialized:
   - `count` is set to 0, which will be used to count consecutive vowels.
   - `flag` is initialized as False, which will be used to indicate if three consecutive vowels are found.
3. The function enters a for loop that iterates through each character `i` in the string `S`.
4. During each iteration, it checks if `i` is in the set of vowels ”aeiou.”
5. In this specific case with S = *”abedfeg”*:
   - The loop encounters ”a” and increments `count` to 1.
   - It then encounters ”b,” which is not a vowel. So, `count` is reset to 0, and `flag` is set to False.
   - It continues with ”e” and increments `count` to 1.

- When it encounters ”d,” which is not a vowel, `count` is reset to 0, and `flag` is set to False.
- The loop proceeds with ”f” and ”g,” which are not vowels. So, `count` remains 0, and `flag` is still False.
- After iterating through all characters in `S`, the code checks `flag`, which remains False.
- The function returns ”Sad” because three consecutive vowels were not found. The result ”Sad” is printed to the console, which is also our expected output

This means the applied logic is correct

6. Game enjoyer

You are playing a video game, and are now fighting the final boss.

The boss has H HP (health points), each normal attack of your character reduces the boss health by X.

Your character also has a special attack that can only be used once during the boss fight, and it will decrease the health of the boss by Y.

You win the battle when the health of the boss is less than 0.

Write a function minattacks that returns the minimum number of attacks needed by your character to win the boss battle.

Make sure to call your function and to present the output use:

print(functionname(parameters))

Constraints

- $1 \leq X < Y \leq H \leq 100$

Interaction

The input comprises a single line containing 3 space-separated integers denoting the values of H, X, Y

The output must contain a number denoting the minimum attacks.

Sample

Input	Output
100 25 40	4
100 29 45	3

In the first case, $(H, X, Y) = (100, 25, 40)$. Your character can attack the boss 4 times normally and that is enough kill the boss. (you can also use special attack and the total attacks would still be 4)

In the second case, $(H, X, Y) = (100, 29, 45)$. Your character can attack the boss 2 times normally and then use the special attack dealing 103 damage in total.

Proposed Solution

```
def minattacks(H,X,Y):
    A = H - Y
    if A % X == 0:
        return A // X + 1
    return A / X + 2
    print(minattacks(X,Y,H))
```

Dry Run
Using one of the inputs provided in the Sample section above, dry run the proposed code solution below.

With $H = 100$, $X = 25$, and $Y = 40$, the code runs as follows:

1. Inside the function, it calculates A as $H - Y$, which is $100 - 40$, resulting in $A = 60$.
2. It checks if A is divisible by X with the condition $A\%X == 0$. In this case, $60\%25$ is not equal to 0, so this condition is False.
3. Since the condition is False, the program resumes from line 5 and returns $A/X + 2$. Since $A = 60$ and $X = 25$, it calculates $60/25 + 2$, which equals $2.4 + 2 = 4.4$.
4. The function will return 4.4, which is not our expected output.
5. The function will also not be called as the function call is unreachable (indented within the function definition)

Error Identification
Briefly explain the errors you identified in the proposed code solution. Mention the line number and the errors in each line.

- Line 5: Floating-point result through simple division (/) rather than floor division (//).
- Line 6: Unreachable function call due to indentation error.

Correct Solution
Rewrite the lines of code you mentioned above with their errors corrected.

```
def minattacks(H,X,Y):
   A = H - Y
   if A % X == 0:
       return A // X + 1
   return A // X + 2
print(minattacks(X,Y,H))
```

Rough Work