Worksheet:

CS 101 Algorithmic Problem Solving

Fall 2023

Name(s): _____

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

1. Odd Boosts

Ahmed plays a game where he gets power boosts on every odd minute. The number of power boosts he gets depends on the odd minute that the boosts are generated at, such that the boosts equal the sum of first 3 multiples of that odd minute.

At the end of every game he wants to figure out the total boosts he got while playing the game for n minutes. Help him find the total boosts given n.

Constraints

- $1 \leq n \leq 99$

Interaction

The input comprises a single line containing an integer denoting the value of n.

The output contains the pattern obtained.

Sample

Input	Output
2	6
5	54

In the first case, he plays the game only for 2 minutes which means he will only get boosts at 1 minute. First 3 multiples of 1 will give 1+2+3 which equals 6.

In the second case, he will get boosts thrice for 1, 3 and 5. so the total will then be the sum of first 5 multiples of 1, 3 and 5 which gives 54 (1+2+3 , 3+6+9 , 5+10+15).

Exercise

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

Input	Output
0	0
1	6
4	24

Problem Identification

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

Given an input *n*, iteratively check a sequence of numbers leading up to n to identify each odd number in the sequence. Once an odd number is identified, iteratively calculate the sum of its first three multiples. Finally, return the sum of the first three multiples for all odd numbers encountered during this process.

Pseudocode

```
n = int(input())
boosts = 0
for i in range(1,n+1):
    if (i%2) != 0:
        for j in range(1,4):
            boosts+= (i*j)
print(boosts)
```

Dry Run

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

Input n=1

i=1
boost= 0
i mod 2!==0 is true so the inner loop executes

var	1st Iter	2nd Iter	3rd Iter
j	1	2	3
boost	0=0+(1*1)=1	1=1+(1*2) = 3	3=3+(1*3)= 6

Output = 6

Input n=4

Outer for loop executes from i=1 to 4

for **i=1***(First Iteration of outer loop)*
boost= 0
i mod 2!==0 is true so the inner loop executes

var	1st Iter	2nd Iter	3rd Iter
j	1	2	3
boost	0=0+(1*1)=1	1=1+(1*2) = 3	3=3+(1*3)= 6

for **i=2***(2nd Iteration of outer loop)*
boost= 6
2 mod 2!==0 is false so the inner loop doesnot executes

for **i=3***(Third Iteration of outer loop)*

boost= 6
3 mod 2!==0 is true so the inner loop executes

var	1st Iter	2nd Iter	3rd Iter
j	1	2	3
boost	6=6+(3*1)=9	9=9+(3*2) = 15	15=15+(3*3)= 24

for **i=4***(4th Iteration of outer loop)*
boost= 24
4 mod 2!==0 is false so the inner loop doesnot executes

Output = 24 which is the expected output

2. Spider Webs

Sarah the spider loves to weave intricate webs. One day, she decided to create a special pattern using her web-weaving skills. She started by weaving concentric circles in her web, with each circle having a different number of threads. The innermost circle had 3 threads, the next had 6 threads, and so on, increasing by 3 threads with each new circle. Sarah wanted to count how many threads she used in total.

If Sarah continues weaving her concentric circles until she has completed n circles in total, and each circle has threads increasing by 3 each time, how many threads will she have used in total?

Note: Solve this using nested loops.

Constraints

- $n \in \mathbb{Z}$
- $1 \leq n \leq 99$

Interaction

The input comprises a single line containing an integer denoting the value of n.

The output contains the total number of threads used.

Sample

Input	Output
3	18
2	9

In the first case, she will make three circles while increasing the threads by 3. So 3+6+9 will give 18.

In the second case, she will make only two circles so 3 and then 6 which will give a total of 9.

Exercise

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

Input	Output
5	45
0	0
1	3

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

Given an input n, iteratively traverse each circle of the web, and within each iteration, iteratively increase the number of weaves by 3. Return the total number of weaves completed.

Pseudocode

```
n = int(input())
sum =0
for i in range(1,n+1):
    for j in range(i*3):
        sum+=1

print(sum)
```

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

For Input n = 0
sum=0
Outer loop will not execute and **Output = 0**

For Input n = 5
sum =0
Outer loop will execute from i=1 to 5

i	j	sum
1	0	1
	1	2
	2	3
2	0	4
	1	5
	2	6
	3	7
	4	8
	5	9
3	0	10
	1	11
	2	12
	3	13
	4	14
	5	15
	6	16
	7	17
	8	18
4	0	19
	1	20
	2	21
	3	22
	4	23
	5	24
	6	25
	7	26
	8	27
	9	28
	10	29
	11	30
5	0	31
	1	32
	2	33
	3	34
	4	35
	5	36
	6	37
	7	38
	8	39
	9	40
	10	41
	11	42
	12	43
	13	44
	14	45

Output= 45 which is the expected Output

3. Numbers and Patterns

Kareem likes to play with numbers and patterns. Given a number n, he wants to make a staircase of numbers for n number of steps (of stairs). Help him develop a program to create such patterns.

Constraints

- $1 \leq n \leq 99$

Interaction

The input comprises a single line containing an integer denoting the value of n.

The output contains the pattern obtained.

Sample

Input	Output
3	1
	1 2
	1 2 3

The above case, keeps counting up the numbers while maintaining the staircase pattern for 3 stairs.

Exercise

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

Input	Output
4	1
	1 2
	1 2 3
	1 2 3 4

Problem Identification

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

Given a number n, incrementally print numbers from 1 to n. Starting with a single *step* containing the number 1, increase the number of steps by 1 in each subsequent iteration, while incrementing the printed numbers on each line. Continue this process until the total number of steps equals n.

Pseudocode

```
n = int(input())
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j, end=" ")
    print()
```

Dry Run

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

Value of 'i'	Values of 'j'
1	1
2	1 2
3	1 2 3
4	1 2 3 4

4. lovely patterns

Sana is an avid enthusiast of intricate patterns and recently stumbled upon a mesmerizing design. Intrigued by its beauty, she decided to write a program that can replicate this enchanting pattern for any given positive integer 'N' using iterative methods.

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

Given the value of N, help Sana print such a pattern using iteration.

Constraints

- $1 \leq N \leq 10$

Interaction

The input contains a single integer value denoting the number of rows the pattern will have

The output would contain multiple lines that show the pattern created.

Sample

Input	Output
3	333
	22
	1
1	1

In the first case, $N = 3$, therefore the first row would contain three 3s, the second would contain two 2s and the final row would just be a 1.

In the second case, $N = 1$, therefore the first row would be the final row and it would contain just a 1.

Exercise

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

Input	Output
2	22
	1
6	6666666
	55555
	4444
	333
	22
	1

Problem Identification

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

Given an input N, use a for loop to print the value of N repeatedly on separate lines. The number of times you print N is determined by a loop variable i. In each iteration of the loop, decrease the value of i by 1, and continue printing N until i reaches 1.

Pseudocode

```
for i in range(N, 0, -1):
    for j in range(i):
        print(i, end=" ")
    print()
```

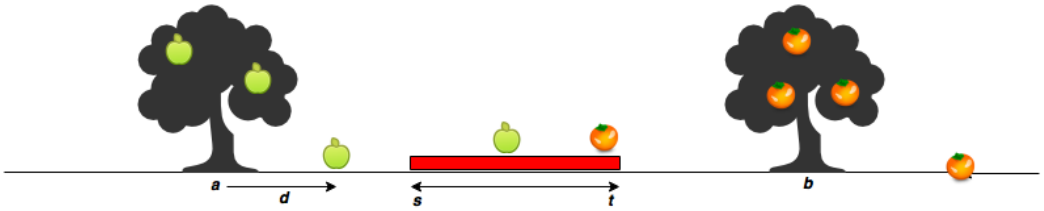
Dry Run

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

Iteration (i)	Inner Loop Iterations (j)
6	6 6 6 6 6 6
5	5 5 5 5 5
4	4 4 4 4
3	3 3 3
2	2 2
1	1

5. Fallen Fruits

Mohammad's house has an apple tree and an orange tree that yield an abundance of fruit. Using the information given below, determine the number of apples and oranges that land on Mohammad's house. (refer to the image provided)



Constraints

- The red shaded region denotes the house, where s is the start point, and t is the endpoint. The apple tree is to the left of the house, and the orange tree is to its right.
- Assume the trees are located on a single point, where the apple tree is at point a, and the orange tree is at point b.
- When a fruit falls from its tree, it lands d units of distance from its tree of origin along the x-axis. A negative value of d means the fruit fell d units to the tree's left, and a positive value of d means it falls d units to the tree's right.

Given the value of d for m apples and n oranges, determine how many apples and oranges will fall on Sam's house (i.e., in the inclusive range $[s, t]$)?

Constraints

- $1 \leq s, t, a, b, m, n \leq 1000$
- $-1000 \leq d \leq 1000$
- $a \leq s \leq t \leq b$

Interaction

The input comprises a single line containing 6 space-separated integers denoting the values of s, t, a, b, m and n respectively. Input has two more lines. The second line contains m space-separated integers denoting the respective distances that each apple falls from point a. The third line contains n space-separated integers denoting the respective distances that each orange falls from point b.

The output must print two space-separated integers. First integer representing the number of apples and second integer representing the number of oranges.

Sample

Input	Output
7 11 5 15 3 2	1 1
-2 2 1	
5 -6	

In the above case: the first apple falls at position (5 - 2) = 3, second falls at (5 + 2) = 7, and the third falls (5 + 1) = 6.

The first orange falls at (15 + 5) = 20 and the second falls at (15 - 6) = 9.

One apple and one orange falls within the range of 7 and 11 hence, the output is 1 and 1.

Exercise

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

Input	Output
8 13 6 18 4 3	2 1
3 -2 2 1	
-4 -6 2	

Problem Identification

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

Given inputs s and t (the range of the house), a (the distance from the apple tree), b (the distance from the orange tree), and m and n (the respective distances at which the apples and oranges fall from their trees), we need to count the number of apples and oranges that fall within the range of the house (between s and t). By taking m distances for falling apples (denoted as A) and n distances for falling oranges (denoted as O), we must check whether the sums of these distances with a and b respectively fall within the range [s, t]. The program increments counters for apples and oranges accordingly and outputs the final counts.

Pseudocode

```
apples, oranges = 0,0
for i in range (m):
    A = int(input())
    if (a + A) >= s and (a + A) <=t:
        apples +=1
```

```
for j in range (n):
    O = int(input())
    if (b + O) >= s and (b + O) <=t:
        oranges +=1
print (apples, oranges)
```

Dry Run

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

Iter.	i	j	Code	apples	oranges
0				0	0
1	0		A = 3	1	0
2	1		A = -2	1	0
3	2		A = 2	2	0
4	3		A = 1	2	0
0				2	0
1		0	O = -4	2	0
2		1	O = -6	2	1
3		2	O = 2	2	1

Output	2 1

6. Secret Code

Ali and his town is under attack, they are waiting for help. The help arrives in form of supplies in the boxes which are labelled with a code each. Ali is skeptical of those boxes but he knows the secret code the attackers use. If the label has an odd number of vowels then it is from the attackers.

Given the label l, figure out if it is safe to open or not.

Constraints

- $1 \leq len(l) \leq 50$

Interaction

The input comprises a single line containing a string that denotes l.

The output must contain a string stating 'safe' or 'not safe' accordingly.

Sample

Input	Output
ahijklo	not safe
kljiom	safe

In the first case, the label 'ahijklo' has 3 vowels (a,i,o). Since the number of vowels is odd (3), hence it is not safe to open the box.

In the second case, the word 'kljiom' has 2 vowels. Since the number of vowels is even, the box is safe to open.

Proposed Solution

```
vowels = 0
for letter in l:
  if l in "aeiou":
    vowels+=1
if vowels % 2 != 0:
  print("safe")
else:
  print("not safe")
```

Dry Run

Using any two of the inputs provided in the Sample section above, dry run the proposed pseudocode in the space below.

Iteration	Letter	Vowels	Condition	Output
1	'a'	1	True	-
2	'h'	1	False	-
3	'i'	1	True	-
4	'j'	1	False	-
5	'k'	1	False	-
6	'l'	1	False	-
7	'o'	1	True	-
8	end of string	1	-	"safe" -(not expected output!)

Error Identification

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

Answer: In line 3, we need to compare each letter (**letter**) not the whole word (**l**).

Conditions and in line 6 and 8 need to be reverse i.e. *not safe* should be printed if **if** condition is executed, and *safe* if **else** condition is executed.

Correct Solution

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

```
vowels = 0
for letter in l:
    if letter in "aeiou":
        vowels+=1
if vowels % 2 == 0:
    print("safe")
else:
    print("not safe")
```

Rough Work