

Bridge Course - Day 03

Submitted By: Afreeen Ahmed

Activity 1: Repetitive Tasks

List three tasks you perform regularly that involve repetition.

For each: 1. What is being repeated?

2. What determines when it stops?

Algorithm & Pseudocode:

Task 1: Sending Emails to a List

1. What is being repeated?

Sending an email to each contact.

2. What determines when it stops?

All contacts in the list are emailed

Create a list of contacts

Algorithm Task 1

For each contact in the list:

a. Send email

Done

Pseudocode:

```
SET contact_list = ["A", "B", "C", ..., "Z"]
```

```
FOR EACH contact IN contact_list
```

```
    PRINT "Sending email to", contact
```

```
END FOR
```

```
PRINT "All emails sent"
```

Task 2: Uploading Assignments

1. What is being repeated?

Uploading one file.

2. What determines when it stops?

All files (e.g., 5) are uploaded.

Algorithm:

```
Set file_number = 1  
While file_number ≤ 5  
    a. Upload the file  
    b. Increment file_number  
Done
```

Pseudocode:

```
SET file_number = 1  
WHILE file_number <= 5  
    PRINT "Uploading file", file_number  
    file_number = file_number + 1  
END WHILE  
PRINT "All assignments uploaded"
```

Task 3: Taking Backup of Files

1. What is being repeated?

Copying each file to backup location.

2. What determines when it stops?

All files are copied.

Algorithm:

Set list of files to back up

For each file in the list:

a. Copy the file to backup folder

Done

Pseudocode:

```
SET files = ["file1", "file2", "file3", ..., "fileN"]  
FOR EACH file IN files  
    PRINT "Backing up", file  
END FOR  
PRINT "Backup completed"
```

Code:

Task 1: Sending Emails to a List

```
contact_list = ["Alice", "Bob", "Charlie", "Diana"]
for contact in contact_list:
    print("Sending email to", contact)
print("All emails sent")
```

Task 2: Uploading Assignments

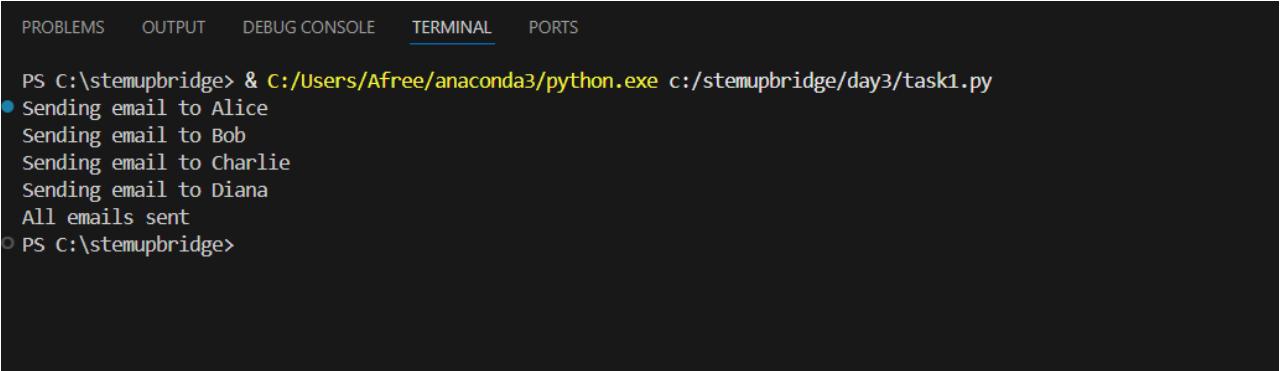
```
file_number = 1
while file_number <= 5:
    print("Uploading file", file_number)
    file_number += 1
print("All assignments uploaded")
```

Task 3: Taking Backup of Files

```
files = ["file1.txt", "file2.txt", "file3.txt", "file4.txt"]
for file in files:
    print("Backing up", file)
print("Backup completed")
```

Output

Case 1:



The screenshot shows a terminal window with the following interface elements at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The main area of the terminal displays the following text:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/task1.py
● Sending email to Alice
● Sending email to Bob
● Sending email to Charlie
● Sending email to Diana
● All emails sent
○ PS C:\stemupbridge>
```

Case 2:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● Sending email to Alice
  Sending email to Bob
  Sending email to Charlie
  Sending email to Diana
  All emails sent
  Uploading file 1
  Uploading file 2
  Uploading file 3
  Uploading file 4
  Uploading file 5
  All assignments uploaded
PS C:\stemupbridge>
```

Case 3:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/task1.py
● Sending email to Alice
  Sending email to Bob
  Sending email to Charlie
  Sending email to Diana
  All emails sent
  Uploading file 1
  Uploading file 2
  Uploading file 3
  Uploading file 4
  Uploading file 5
  All assignments uploaded
  Backing up file1.txt
  Backing up file2.txt
  Backing up file3.txt
  Backing up file4.txt
  Backup completed
PS C:\stemupbridge>
```

Activity 2: Code Duplication

Write how you would print “Hello!” 10 times without loops. Reflect on how loops make this easier for 1000 times

Algorithm:

1. Print "Hello!"
 2. Repeat the print statement 10 times manually
 3. End

Pseudocode:

Code:

- Without loops, we have to **repeat the same line manually**.
 - For 1000 times, it would mean **writing 1000 lines**, which is:

1. Time-consuming
 2. Hard to read and maintain
 3. Error-prone
- **Loops simplify repetition** by letting you write a **few lines** that do the task multiple times automatically, saving time and effort

Algorithm (With Loop):

1. Set counter = 1
2. While counter \leq 1000
 - a. Print "Hello!"
 - b. Increment counter
3. End

Pseudocode (With Loop):

```
SET counter = 1
WHILE counter <= 1000
    PRINT "Hello!"
    counter = counter + 1
END WHILE
```

Problem 3: Countdown Print numbers from 10 to 1, then “Blastoff!”

Algorithm:

1. Set a counter to 10.
2. While the counter is greater than or equal to 1:
 - a. Print the counter.
 - b. Decrease the counter by 1.
3. After the loop ends, print "Blastoff!"

Pseudocode:

```
SET counter TO 10
WHILE counter >= 1 DO
    PRINT counter
    counter = counter - 1
END WHILE
PRINT "Blastoff!"
```

Code:

```
counter = 10
while counter >= 1:
    print(counter)
    counter -= 1
print("Blastoff!")
```

Output:

Case 1:

A screenshot of a terminal window from a code editor. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal output shows a countdown from 10 to 1, followed by the text "Blastoff!". The command entered was "python c:/stemupbridge/day3/counter.py".

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● 10
9
8
7
6
5
4
3
2
1
Blastoff!
○ PS C:\stemupbridge>
```

Case 2:

A screenshot of a terminal window from a code editor. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal output shows a syntax error: "PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py". It then shows a countdown from 5 to 1, followed by the text "Blastoff!". The command entered was "& C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py".

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py
● 5
4
3
2
1
Blastoff!
○ PS C:\stemupbridge>
```

Case 3:

A screenshot of a terminal window from a code editor. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal output shows a syntax error: "PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py". It then shows a syntax error: "● Blastoff!". The command entered was "& C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py".

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/counter.py
● Blastoff!
○ PS C:\stemupbridge>
```

Problem 4: Sum Until Zero

Ask user for numbers repeatedly until they enter 0. Sum and print the total.

Algorithm:

1. Initialize sum to 0.
2. Ask the user to enter a number.
3. While the number is not 0:
 - a. Add the number to sum.
 - b. Ask for the next number.
4. When 0 is entered, print the total sum.

Pseudocode:

```
SET sum TO 0
REPEAT
    PROMPT user for number
    READ number
    sum = sum + number
UNTIL number == 0
PRINT sum
```

Code:

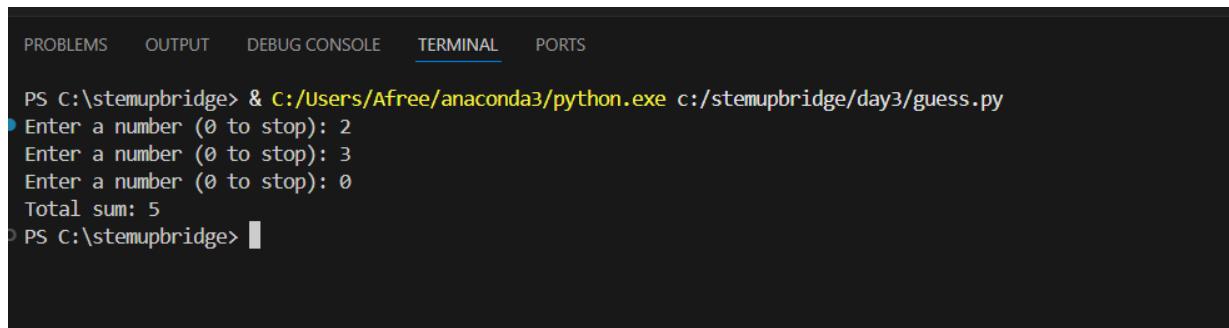
```
total = 0
while True:
    num = int(input("Enter a number (0 to stop): "))
    if num == 0:
        break
    total += num
print("Total sum:", total)
```

Output:

Case 1: Given 4 number as input

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/guess.py
● Enter a number (0 to stop): 6
Enter a number (0 to stop): 7
Enter a number (0 to stop): 8
Enter a number (0 to stop): 0
Total sum: 21
○ PS C:\stemupbridge>
```

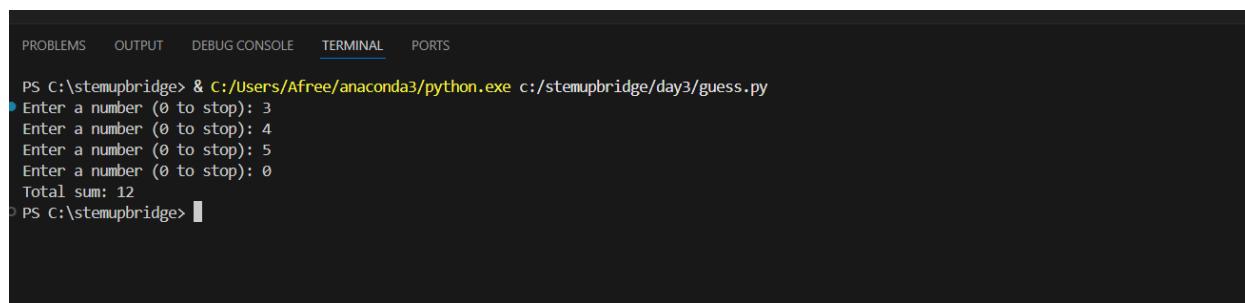
Case 2: Given 2 number as input



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/guess.py
> Enter a number (0 to stop): 2
> Enter a number (0 to stop): 3
> Enter a number (0 to stop): 0
> Total sum: 5
> PS C:\stemupbridge>
```

Case 3: Given 3 number as input



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/guess.py
> Enter a number (0 to stop): 3
> Enter a number (0 to stop): 4
> Enter a number (0 to stop): 5
> Enter a number (0 to stop): 0
> Total sum: 12
> PS C:\stemupbridge>
```

Problem 5: Guess the Number

Generate a random number between 1 and 10. Ask user to guess. Provide feedback and loop until correct.

Algorithm:

1. Set the secret number (e.g., 7).
2. Ask the user to guess the number.
3. While the guess is not equal to the secret number:
 - a. If guess > secret, print "Too high".
 - b. If guess < secret, print "Too low".
 - c. Ask again.
4. When guess is correct, print "Correct!".

Pseudocode:

```
SET secret TO 7
PROMPT user to guess the number
READ guess
WHILE guess ≠ secret DO
    IF guess > secret THEN
        PRINT "Too high"
    ELSE
        PRINT "Too low"
    END IF
    PROMPT user to guess again
    READ guess
END WHILE
PRINT "Correct!"
```

Code:

```
import random

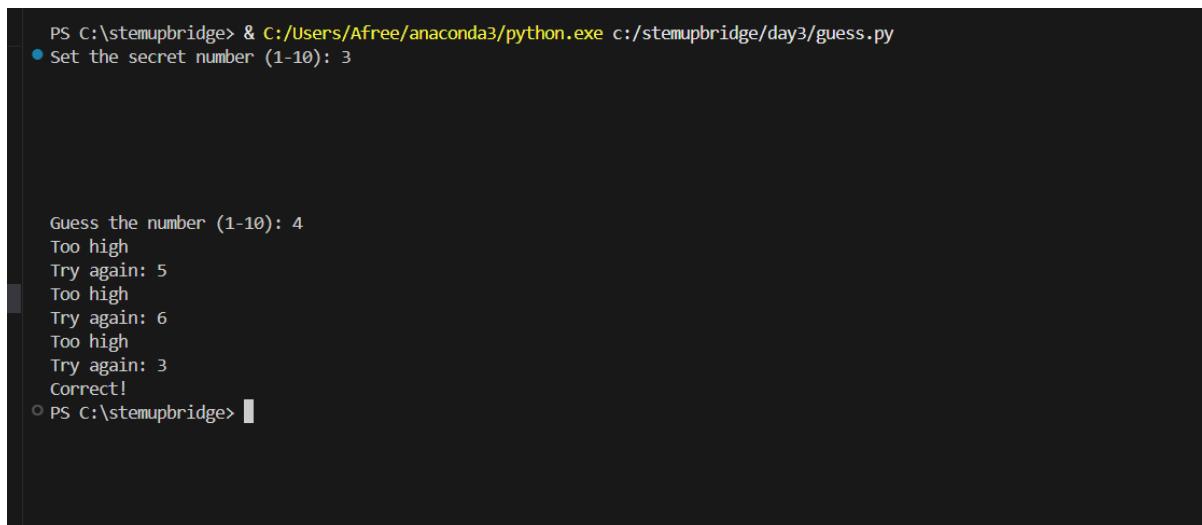
a = random.randint(1, 10)
print("\n" * 5)

guess = int(input("Guess the number (1-10): "))

while guess != a:
    if guess > a:
        print("Too high")
    else:
        print("Too low")
    guess = int(input("Try again: "))

print("Correct!")
```

Output:



```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/guess.py
● Set the secret number (1-10): 3

Guess the number (1-10): 4
Too high
Try again: 5
Too high
Try again: 6
Too high
Try again: 3
Correct!
○ PS C:\stemupbridge>
```

Problem 6: Infinite Loop Debugging Analyze and fix

```
int counter = 0;
while (counter < 5) {
    System.out.println("Hello");
}
```

Issue in Code:

```
while (counter < 5) {
    System.out.println("Hello");
}
```

It will run forever because counter is never incremented, so counter < 5 is always true.

Algorithm:

Initialize counter = 0

While counter < 5:

- a. Print "Hello"
- b. Increment counter

End

Pseudocode :

```
SET counter = 0
WHILE counter < 5
    PRINT "Hello"
    counter = counter + 1
END WHILE
```

Code:

```
int counter = 0;
while (counter < 5) {
    System.out.println("Hello");
    counter++; // Fix: increment counter to avoid infinite loop
}
```

Output

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stempupbridge> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Afree\A
● bdb22434a24fdf6cac771e5b56ca\redhat.java\jdt_ws\stempupbridge_359bee65\bin' 'Main'
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
○ PS C:\stempupbridge>
```

Problem 7: Even Numbers Print even numbers from 2 to 20 using a for loop.

Algorithm:

1. Loop from 2 to 20
2. Step by 2
3. Print each number

Pseudocode:

```
FOR i = 2 TO 20 STEP 2
    PRINT i
END FOR
```

Code:

```
for i in range(2, 21):
    if i % 2 == 0:
        print(i)
```

Output:

```
● PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/even.py
2
4
6
8
10
12
14
16
18
20
○ PS C:\stemupbridge> █
```

Problem 8: Factorial Calculator Calculate n! for user input n. Handle edge case when n == 0.

Algorithm:

1. Input n
2. If n == 0 → print 1
3. Else: Loop from 1 to n
 - a. Multiply values into fact
4. Print fact

Pseudocode:

```
INPUT n
IF n == 0 THEN
    PRINT 1
ELSE
    SET fact = 1
    FOR i = 1 TO n
        fact = fact * i
    END FOR
    PRINT fact
END IF
```

Code:

```
n = int(input("Enter a number: "))
if n == 0:
    print("Factorial: 1")
else:
    fact = 1
    for i in range(1, n+1):
        fact *= i
    print("Factorial:", fact)
```

Output:

Problem 9: Count ‘a’ in String Ask for a string input. Count how many times ‘a’ or ‘A’ appears.

Algorithm:

1. Input a string
2. Set count = 0
3. For each character
4. If char is 'a' or 'A',
5. increment count
6. Print count

Pseudocode:

```
INPUT text  
SET count = 0  
FOR each character IN text  
    IF character == 'a' OR character == 'A' THEN  
        count = count + 1  
    END IF  
END FOR  
PRINT count
```

Code:

```
text = input("Enter a string: ")  
count = 0  
for char in text:  
    if char == 'a' or char == 'A':  
        count += 1  
print("Count of 'a' or 'A':", count)
```

Output:

The screenshot shows a terminal window with the following interface elements at the top:

- PROBLEMS 1
- OUTPUT
- DEBUG CONSOLE
- TERMINAL
- PORTS

The terminal window displays the following text:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/coutn.py
● Enter a string: asdfghaaaaaaaaaa
Count of 'a' or 'A': 8
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/coutn.py
● Enter a string:aaaaaaaaaaaaaaaaaa
Count of 'a' or 'A': 15
● PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/coutn.py
Enter a string: sdfghjkl
Count of 'a' or 'A': 0
○ PS C:\stemupbridge> █
```

Problem 10: Simple Star Pattern Print: ***** Using one for loop.

Algorithm:

1. Loop from 1 to 5
2. Print "*" in one line

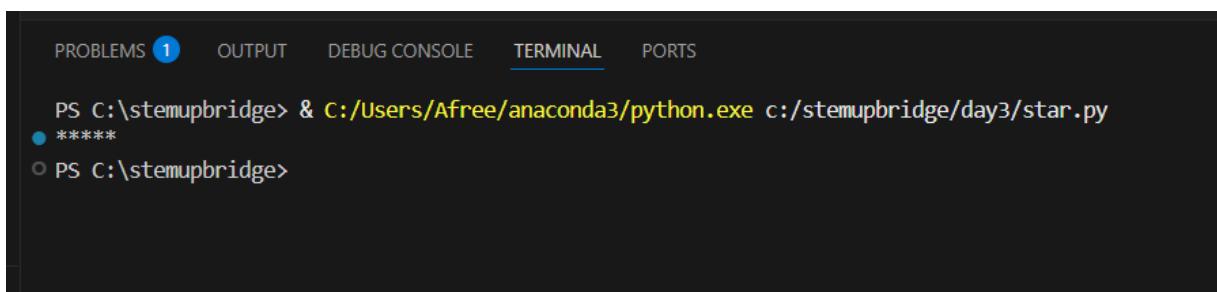
Pseudocode:

```
FOR i = 1 TO 5
    PRINT "*" without newline
END FOR
```

Code:

```
for i in range(5):
    print("*", end="")
print()
```

Output:



The screenshot shows a terminal window with the following interface elements at the top: PROBLEMS (1), OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS. The terminal content is as follows:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/star.py
*****
PS C:\stemupbridge>
```

Problem 3.1: Prime Checker

Check if a number is prime using a loop and break.

Algorithm:

1. Input a number n
2. If $n < 2$, it is not prime
3. Check divisibility from 2 to \sqrt{n}
4. If divisible \rightarrow not prime (break)
5. Else \rightarrow prime

Pseudocode:

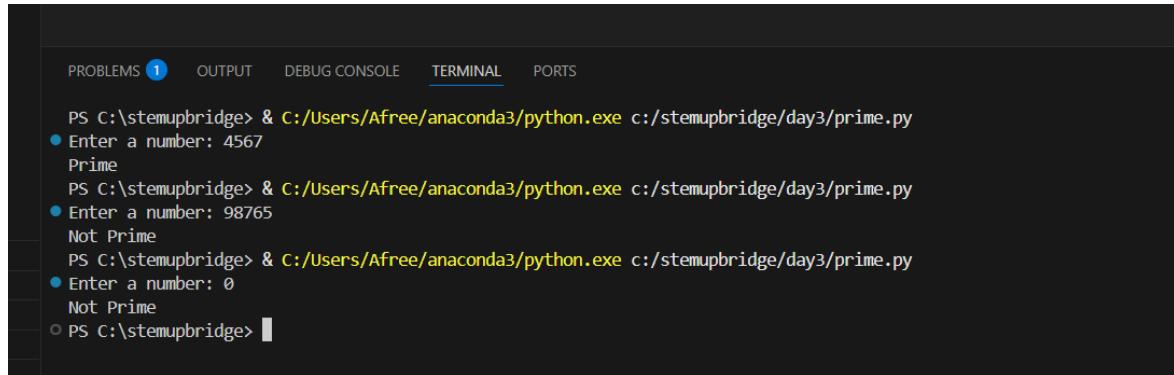
```
INPUT n
IF n < 2 THEN
    PRINT "Not Prime"
ELSE
    SET is_prime = True
    FOR i = 2 TO (n // 2)
        IF n MOD i == 0 THEN
            SET is_prime = False
            BREAK
        END IF
    END FOR
    IF is_prime THEN
        PRINT "Prime"
    ELSE
        PRINT "Not Prime"
    END IF
END IF
```

Code:

```
n = int(input("Enter a number: "))
if n < 2:
    print("Not Prime")
else:
    for i in range(2, n):
        if n % i == 0:
```

```
print("Not Prime")
break
else:
    print("Prime")
```

Output:



The screenshot shows a terminal window with the following text output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/prime.py
● Enter a number: 4567
Prime
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/prime.py
● Enter a number: 98765
Not Prime
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/prime.py
● Enter a number: 0
Not Prime
○ PS C:\stemupbridge> █
```

Problem 3.2: Skip Negatives Input 5 numbers. Use continue to skip negative ones and sum the rest.

Algorithm:

1. Repeat 5 times:
 - a. Input number
 - b. If number is negative → skip (continue)
 - c. Else → add to sum
2. Print sum

Pseudocode:

```
SET sum = 0
FOR i = 1 TO 5
    INPUT num
    IF num < 0 THEN
        CONTINUE
    ELSE
        sum = sum + num
    END IF
END FOR
PRINT sum
```

Code:

```
total = 0
for i in range(5):
    num = int(input("Enter a number: "))
    if num < 0:
        continue
    total += num
print("Sum of non-negative numbers:", total)
```

Output:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/nedative.py
● Enter a number: 67
Enter a number: -987
Enter a number: 87
Enter a number: 76
Enter a number: 67
Sum of non-negative numbers: 297
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/nedative.py
● Enter a number: 3456
Enter a number: -0987
Enter a number: 567
Enter a number: -098
Enter a number: 45678
Sum of non-negative numbers: 49701
● PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/nedative.py
Enter a number: 3456
Enter a number: 3456
Enter a number: 3456
Enter a number: -567
Enter a number: 5678
Sum of non-negative numbers: 16046
○ PS C:\stemupbridge> █
```

Problem 3.3: Rectangle Pattern Input rows and cols, print a rectangle of *.

Algorithm:

1. Input rows and cols
2. For each row
3. Print cols stars

Pseudocode:

```
INPUT rows, cols  
FOR i = 1 TO rows  
    FOR j = 1 TO cols  
        PRINT "*", no newline  
    END FOR  
    PRINT newline  
END FOR
```

Code:

```
rows = int(input("Enter rows: "))  
cols = int(input("Enter columns: "))  
for i in range(rows):  
    for j in range(cols):  
        print("*", end="")  
    print()
```

Output:

```
PS C:\stemupbridge> & c:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3
● Enter length: 8
Enter breath: 4
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
```

Problem 3.4: Triangle Pattern Input height. Print right-angled triangle with *.

Algorithm:

1. Input height
 2. Loop from 1 to height
 3. Print i stars in each row

Pseudocode:

```

INPUT height
FOR i = 1 TO height
    PRINT "*" i times
END FOR

```

Code:

```
height = int(input("Enter height: "))

for i in range(1, height + 1):
    print("*" * i)
```

Output:

```
PS C:\stemupbridge> & C:/Users/Afree/anaconda3/python.exe c:/stemupbridge/day3/triangle.py
● Enter height: 12
*
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
* * * * * * * * * * *
```

Problem 3.5: Pyramid Pattern Challenge Input height. Print centered pyramid:

Algorithm:

1. Input height
2. For each level from 1 to height
 - a. Print spaces: height - i
 - b. Print stars: $2*i - 1$

Pseudocode:

```
INPUT height
FOR i = 1 TO height
    PRINT (height - i) spaces
    PRINT (2*i - 1) stars
    PRINT newline
END FOR
```

Code:

```
height = int(input("Enter height: "))
for i in range(height):
    for space in range(height - i - 1):
```

```
print(" ", end="") # Print leading spaces  
for star in range(2 * i + 1):  
    print("*", end="") # Print stars  
print() # Move to next line
```

Output