

# Rajalakshmi Engineering College

Name: Adithya S  
Email: 240701019@rajalakshmi.edu.in  
Roll no: 240701019  
Phone: 9840784531  
Branch: REC  
Department: I CSE FA  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

Taylor is tasked with a mathematical challenge that requires finding the smallest positive number divisible by all integers from 1 to n.

Help Taylor to determine the smallest positive number that is divisible by all integers from 1 to n. Make sure to employ the break statement to ensure efficiency in the program.

#### ***Input Format***

The input consists of a single integer, n.

#### ***Output Format***

The output displays the smallest positive number that is divisible by all integers from 1 to n.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 10

Output: 2520

### **Answer**

```
# You are using Python
import math
def l(a,b):
    return abs(a*b)//math.gcd(a,b)
n=int(input())
r=1
for i in range(1,n+1):
    r=l(r,i)
print(r)
```

**Status :** Correct

**Marks : 10/10**

## **2. Problem Statement**

John is tasked with configuring the lighting for a high-profile event, where different lighting modes affect the ambiance of the venue. He can choose from three distinct lighting modes, each requiring a specific adjustment to the initial light intensity:

Ambient Lighting (Mode 1): The intensity level is multiplied by 1.5. Stage Lighting (Mode 2): The intensity level is multiplied by 2.0. Spotlight (Mode 3): The intensity level is multiplied by 1.8.

In the event that an invalid mode is provided, the program should output an error message indicating the invalid selection.

Your task is to write a program that reads the selected lighting mode and

the initial intensity level, applies the appropriate adjustment, and prints the final intensity.

### ***Input Format***

The first line of input is an integer  $n$ , representing the lighting mode.

The second line is a floating value  $m$ , representing the initial intensity level of the light.

### ***Output Format***

The output displays "Intensity: " followed by a float representing the adjusted intensity level, formatted to two decimal places, if the mode is valid.

If the mode is invalid, the output should display "Invalid".

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 1

10.0

Output: Intensity: 15.00

### ***Answer***

# You are using Python

```
n=int(input())
```

```
m=float(input())
```

```
if(n==1):
```

```
    d=m*1.5
```

```
    print(f"Intensity: {d:.2f}")
```

```
elif(n==2):
```

```
    d=m*2
```

```
    print(f"Intensity: {d:.2f}")
```

```
elif(n==3):
```

```
    d=m*1.8
```

```
    print(f"Intensity: {d:.2f}")
```

```
else:
```

```
    print("Invalid")
```

Status : Correct

Marks : 10/10

### 3. Problem Statement

Rohith is a data analyst who needs to categorize countries based on their population growth rates. Each country is assigned a unique code. Rohith will receive a code and corresponding data based on the code. If the data falls within specific thresholds, he needs to classify the country's priority level.

Your task is to write a program that reads a country code and its associated data, and then determines if the priority is "High" or "Low."

Thresholds: France: Priority is "High" if the percentage < 50, else "Low". Japan: Priority is "High" if life expectancy > 80, else "Low". Brazil: Priority is "High" if the urban population > 80, else "Low".

#### **Input Format**

The first line of input consists of an integer, representing the country code (1 for France, 2 for Japan, 3 for Brazil).

If the country code is 1,

- The second line consists of a floating-point value N, representing the percentage of the English-speaking population.

If the country code is 2,

- The second line consists of a floating-point value A, representing the average life expectancy in years.

If the country code is 3,

- The second line consists of a floating-point value P, representing the percentage of the urban population.

#### **Output Format**

The first line of output displays "Priority: High" or "Priority: Low" based on the input data.

If the country code is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1

30.0

Output: Priority: High

### **Answer**

# You are using Python

```
import sys
```

```
n=int(input())
```

```
if(n>3):
```

```
    print("Invalid")
```

```
    exit(0)
```

```
m=float(input())
```

```
a="High"
```

```
b="Low"
```

```
c="Priority:"
```

```
if(n==1):
```

```
    if(m<50):
```

```
        print(f"Priority: {a}")
```

```
    else:
```

```
        print(f"Priority: {b}")
```

```
elif(n==2):
```

```
    if(m>80):
```

```
        print(f"{c} {a}")
```

```
    else:
```

```
        print(f"{c} {b}")
```

```
elif(n==3):
```

```
    if(m>80):
```

```
        print(f"{c} {a}")
```

```
    else:
```

```
        print(f"{c} {b}")
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Max is fascinated by prime numbers and the Fibonacci sequence. He wants to combine these two interests by creating a program that outputs the first  $n$  prime numbers within the Fibonacci sequence.

Your task is to help Max by writing a program that prints the first  $n$  prime numbers in the Fibonacci sequence using a while loop along with the break statement to achieve the desired functionality.

##### ***Input Format***

The input consists of an integer  $n$ , representing the number of prime Fibonacci numbers to generate.

##### ***Output Format***

The output displays space-separated first  $n$  prime numbers found in the Fibonacci sequence.

Refer to the sample output for the formatting specifications.

##### ***Sample Test Case***

Input: 5

Output: 2 3 5 13 89

##### ***Answer***

```
# You are using Python
n=int(input())
a,b=0,1
m=[]
def p(n):
    if n<2:
        return False
    for i in range(2,int(n**.5)+1):
        if n%i==0:
            return False
    return True
while(n>0):
```

```
c=a+b
a,b=b,c
if(p(c)==True):
    m.append(c)
    n-=1
for i in range(len(m)):
    print(m[i],end=" ")
```

**Status :** Correct

**Marks :** 10/10