

# Rajalakshmi Engineering College

Name: Jenell S G  
Email: 240701212@rajalakshmi.edu.in  
Roll no: 2116240701212  
Phone: 7418493255  
Branch: REC  
Department: I CSE AH  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_PAH

Attempt : 1  
Total Mark : 60  
Marks Obtained : 60

### Section 1 : Coding

#### 1. Problem Statement

Tom wants to create a dictionary that lists the first n prime numbers, where each key represents the position of the prime number, and the value is the prime number itself.

Help Tom generate this dictionary based on the input she provides.

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

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

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

The output displays the generated dictionary where each key is an integer from 1 to n, and the corresponding value is the prime number.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

Output: {1: 2, 2: 3, 3: 5, 4: 7}

### **Answer**

# You are using Python

```
def is_prime(num):  
    """Check if a number is prime."""  
    if num < 2:  
        return False  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            return False  
    return True  
  
def generate_primes(n):  
    """Generate a dictionary of the first n prime numbers."""  
    primes = {}  
    count = 0  
    num = 2 # Start checking for prime numbers from 2  
  
    while count < n:  
        if is_prime(num):  
            count += 1  
            primes[count] = num  
            num += 1  
  
    return primes  
  
def main():  
    # Read the input value for n  
    n = int(input().strip())  
  
    # Generate the dictionary of prime numbers  
    prime_dict = generate_primes(n)  
  
    # Print the resulting dictionary
```

```
print(prime_dict)
```

```
# Call the main function to execute the program  
main()
```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Sophia is organizing a list of event IDs representing consecutive days of an event. She needs to group these IDs into consecutive sequences. For example, if the IDs 3, 4, and 5 appear consecutively, they should be grouped.

Write a program that helps Sophia by reading the total number of event IDs and the IDs themselves, then display each group of consecutive IDs in tuple format.

### ***Input Format***

The first line of input consists of an integer  $n$ , representing the number of event IDs.

The next  $n$  lines contain integers representing the event IDs, where each integer corresponds to an event ID.

### ***Output Format***

The output should display each group of consecutive event IDs in a tuple format. Each group should be printed on a new line, and single event IDs should be displayed as a single-element tuple.

Refer to the sample output for formatting specifications.

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

Input: 3

1

2

3

Output: (1, 2, 3)

### **Answer**

```
# Read the number of event IDs
n = int(input())

# Read the event IDs
event_ids = [int(input()) for _ in range(n)]

# Sort the event IDs to ensure they are in order
event_ids.sort()

# Initialize a list to store the groups of consecutive event IDs
groups = []
current_group = [event_ids[0]]

# Group consecutive IDs
for i in range(1, n):
    if event_ids[i] == event_ids[i - 1] + 1:
        current_group.append(event_ids[i])
    else:
        groups.append(tuple(current_group))
        current_group = [event_ids[i]]

# Append the last group
groups.append(tuple(current_group))

# Print each group as a tuple on the same line
for i, current_group in enumerate(groups):
    if len(current_group) == 1:
        print(f"({current_group[0]})", end = " ")
    else:
        print(current_group, end = " ")
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Mia is organizing a list of integers into a series of pairs for his new project. She wants to create pairs of consecutive integers from the list. The last

integer should be paired with None to complete the series. The pairing happens as follows: ((Element 1, Element 2), (Element 2, Element 3)..... (Element n, None)).

Your task is to help Henry by writing a Python program that reads a list of integers, forms these pairs, and displays the result in tuple format.

### ***Input Format***

The first line of input consists of an integer n, representing the number of elements in the tuple.

The second line of input contains n space-separated integers, representing the elements of the tuple.

### ***Output Format***

The output displays a tuple containing pairs of consecutive integers from the input. The last integer in the tuple is paired with 'None'.

Refer to the sample output for formatting specifications.

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

Input: 3

5 10 15

Output: ((5, 10), (10, 15), (15, None))

### ***Answer***

```
# You are using Python
```

```
def create_pairs():
```

```
    # Read the number of elements
```

```
    n = int(input().strip())
```

```
    # Read the elements into a list
```

```
    elements = list(map(int, input().strip().split()))
```

```
    # Initialize a list to hold the pairs
```

```
    pairs = []
```

```
    # Create pairs of consecutive integers
```

```
for i in range(n - 1):
    pairs.append((elements[i], elements[i + 1]))

# Add the last pair with None
pairs.append((elements[-1], None))

# Convert the list of pairs to a tuple
result = tuple(pairs)

# Print the result
print(result)

# Call the function to execute the program
create_pairs()
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Jordan is creating a program to process a list of integers. The program should take a list of integers as input, remove any duplicate integers while preserving their original order, concatenate the remaining unique integers into a single string, and then print the result.

Help Jordan in implementing the same.

##### **Input Format**

The input consists of space-separated integers representing the elements of the set.

##### **Output Format**

The output prints a single integer formed by concatenating the unique integers from the input in the order they appeared.

Refer to the sample output for the formatting specifications.

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

Input: 11 11 33 50

Output: 113350

### **Answer**

# You are using Python

```
def process_integers():
```

```
    # Read the input as a list of integers
```

```
    input_numbers = input().strip().split()
```

```
    # Initialize a list to hold unique integers
```

```
    unique_numbers = []
```

```
    seen = set() # To track seen integers
```

```
    for number in input_numbers:
```

```
        if number not in seen:
```

```
            unique_numbers.append(number) # Add to unique list
```

```
            seen.add(number) # Mark as seen
```

```
    # Concatenate the unique integers into a single string
```

```
    result = "".join(unique_numbers)
```

```
    # Print the result
```

```
    print(result)
```

```
# Call the function to execute the program
```

```
process_integers()
```

**Status : Correct**

**Marks : 10/10**

## **5. Problem Statement**

Rishi is working on a program to manipulate a set of integers. The program should allow users to perform the following operations:

Find the maximum value in the set. Find the minimum value in the set. Remove a specific number from the set.

The program should handle these operations based on user input. If the user inputs an invalid operation choice, the program should indicate that the choice is invalid.

### ***Input Format***

The first line contains space-separated integers that will form the initial set. Each integer  $x$  is separated by a space.

The second line contains an integer  $ch$ , representing the user's choice:

- 1 to find the maximum value
- 2 to find the minimum value
- 3 to remove a specific number from the set

If  $ch$  is 3, the third line contains an integer  $n1$ , which is the number to be removed from the set.

### ***Output Format***

The first line of output prints the original set in descending order.

For choice 1: Print the maximum value from the set.

For choice 2: Print the minimum value from the set.

For choice 3: Print the set after removing the specified number, in descending order.

For invalid choices: Print "Invalid choice".

Refer to the sample output for the formatting specifications.

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

Input: 1 2 3 4 5

1

Output: {5, 4, 3, 2, 1}

5

### ***Answer***

```
# You are using Python
# Step 1: Read and parse the input set
initial_input = input().strip()
numbers = set(map(int, initial_input.split()))
```



# Step 2: Read the choice

try:

ch = int(input().strip())

except:

ch = None # Invalid input for choice

# Step 3: Print the original set in descending order

sorted\_set = sorted(numbers, reverse=True)

print("{ " + ", ".join(map(str, sorted\_set)) + "}")

# Step 4: Handle operations based on the choice

if ch == 1:

print(max(numbers))

elif ch == 2:

print(min(numbers))

elif ch == 3:

try:

n1 = int(input().strip())

numbers.discard(n1) # Discard safely removes if present

sorted\_set = sorted(numbers, reverse=True)

print("{ " + ", ".join(map(str, sorted\_set)) + "}")

except:

print("Invalid choice")

else:

print("Invalid choice")

**Status :** Correct

**Marks :** 10/10

## 6. Problem Statement

Maya wants to create a dictionary that maps each integer from 1 to a given number  $n$  to its square. She will use this dictionary to quickly reference the square of any number up to  $n$ .

Help Maya generate this dictionary based on the input she provides.

### Input Format

The input consists of an integer  $n$ , representing the highest number for which Maya wants to calculate the square.

### **Output Format**

The output displays the generated dictionary where each key is an integer from 1 to n, and the corresponding value is its square.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 5

Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

### **Answer**

```
# You are using Python
```

```
# Read input
```

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

```
# Generate dictionary
```

```
squares_dict = {i: i**2 for i in range(1, n + 1)}
```

```
# Print output
```

```
print(squares_dict)
```

**Status :** Correct

**Marks : 10/10**