Rajalakshmi Engineering College

Name: ANIRUDH R
Email: 240701039@rajalakshmi.edu.in
Roll no: 240701039
Phone: 9363540767
Branch: REC
Department: I CSE FA
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 : 46

Section 1 : Coding

1. 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

```
n=int(input())
e=list(map(int,input().split()))
p=[]
for i in range(n - 1):
    p.append((e[i],e[i+1]))
p.append((e[-1], None))
r=tuple(p)
print(r)
```

Status : Correct ***Marks : 10/10***

2. 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

```
n=int(input())
s={i:i**2 for i in range(1,n+1)}
print(s)
```

Status : Correct ***Marks : 10/10***

3. 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

```
input_numbers = input().split()
unique_numbers = []
seen = set()
for number in input_numbers:
    if number not in seen:
        unique_numbers.append(number)
        seen.add(number)
result = ''.join(unique_numbers)
print(result)
```

Status : Correct ***Marks : 10/10***

4. 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

```
n = int(input())
primes = []
num = 2
while len(primes) < n:
    is_prime = True
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            is_prime = False
            break
    if is_prime:
        primes.append(num)
    num += 1
prime_dict = {i + 1: primes[i] for i in range(n)}
print(prime_dict)
```

Status : Correct ***Marks : 10/10***

5. 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

```
n = int(input())
event_ids = []
for _ in range(n):
    event_ids.append(int(input()))
consecutive_groups = []
current_group = []
for i in range(len(event_ids)):
    if i == 0 or event_ids[i] == event_ids[i - 1] + 1:
        current_group.append(event_ids[i])
    else:
        consecutive_groups.append(tuple(current_group))
        current_group = [event_ids[i]]
if current_group:
    consecutive_groups.append(tuple(current_group))
for group in consecutive_groups:
    print(group)
```

Status : Partially correct ***Marks : 6/10***

6. 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

```
# Read the initial set of integers
initial_set = list(map(int, input().split()))

# Convert to a set to ensure uniqueness, then back to a sorted list in descending
order
unique_set = sorted(set(initial_set), reverse=True)

# Print the original set in descending order
print(set(unique_set))

# Read the user's choice
choice = int(input())

# Perform actions based on user choice
if choice == 1:
    # Find and print the maximum value
    max_value = max(unique_set)
    print(max_value)
elif choice == 2:
    # Find and print the minimum value
    min_value = min(unique_set)
    print(min_value)
elif choice == 3:
    # Read the number to remove
    number_to_remove = int(input())
    if number_to_remove in unique_set:
        unique_set.remove(number_to_remove)
        # Print the updated set in descending order
        print(set(sorted(unique_set, reverse=True)))
    else:
        print(f"{number_to_remove} not found in the set.")
else:
    # Handle invalid choice
    print("Invalid choice")
```

Status : Wrong ***Marks : 0/10***