

1. Створити програму для генерації довільного неспадаючого масиву із 100 цілих чисел, які не перевищують по модулю 255, з наступними властивостями: 1.1. При кожному запуску програми масив повинен бути різним. 1.2. Масив повинен містити 10 елементів з однаковим значенням a в довільному місці (а випадкове значення із $[1, 254]$). 1.3. Масив повинен містити 20 елементів з однаковим значенням b в довільному місці ($b \neq a$, випадкове значення із $[1, 254]$). 1.4. a та b задає користувач.

```
In [24]: import random

def generate_array(a, b):
    if a == b or not (1 <= a <= 254) or not (1 <= b <= 254):
        raise ValueError("a і b повинні бути різними та належати діапазону [1, 254]")

    # Генеруємо 70 випадкових чисел із діапазону [-255, 255] виключаючи значення
    random_numbers = [random.choice([i for i in range(1, 256) if i != a and i != b]) for _ in range(70)]

    # Додаємо 10 чисел a і 20 чисел b
    array = random_numbers + [a] * 10 + [b] * 20

    # Перемішуємо масив
    random.shuffle(array)

    # Сортуємо масив за неспаданням
    array.sort()
    random.shuffle(array)

    return array

# Введення значень a та b від користувача
# a = int(input("Введіть значення a (від 1 до 254): "))
# b = int(input("Введіть значення b (від 1 до 254): "))

a = 10
b = 80

# Генерація масиву
result_array = generate_array(a, b)

# Виведення результату
print("Згенерований масив:", result_array)
```

Згенерований масив: [10, 63, 130, 80, 80, 80, 80, 10, 10, 10, 81, 153, 185, 171, 10, 80, 184, 126, 91, 112, 80, 180, 101, 82, 20, 14, 146, 80, 80, 201, 10, 85, 80, 224, 242, 1, 185, 72, 15, 162, 80, 170, 80, 67, 150, 80, 39, 74, 80, 237, 236, 80, 103, 218, 224, 235, 10, 71, 233, 10, 167, 82, 166, 22, 26, 80, 173, 73, 30, 22, 80, 106, 143, 127, 10, 80, 80, 199, 10, 164, 147, 113, 11, 37, 146, 120, 83, 151, 97, 252, 75, 80, 80, 131, 138, 236, 179, 77, 215, 250]

Зберегти згенерований масив в файл.

```
In [25]: def save_array_to_file(array, filename):
    # Записуємо масив у файл
    with open(filename, 'w') as file:
        for num in array:
```

```

        file.write(f"{num}\n")

# Зберігаємо масив у файл
save_array_to_file(result_array, 'output_array.txt')
print(f"Масив збережено у файл 'output_array.txt'")

```

Масив збережено у файл 'output_array.txt'

3. Створити програму, яка зчитує із файлу згенерований масив попередньою програмою і виводить його на екран.

```

In [26]: def read_array_from_file(filename):
# Читаємо масив з файлу
with open(filename, 'r') as file:
    array = [int(line.strip()) for line in file]
    return array

# Ім'я файлу, з якого будемо читати масив
filename = 'output_array.txt'

# Читаємо масив з файлу
array_from_file = read_array_from_file(filename)

# Виводимо зчитаний масив на екран
print("Зчитаний масив з файлу:", array_from_file)

```

Зчитаний масив з файлу: [10, 63, 130, 80, 80, 80, 80, 10, 10, 10, 81, 153, 185, 171, 10, 80, 184, 126, 91, 112, 80, 180, 101, 82, 20, 14, 146, 80, 80, 201, 10, 85, 80, 224, 242, 1, 185, 72, 15, 162, 80, 170, 80, 67, 150, 80, 39, 74, 80, 237, 236, 80, 103, 218, 224, 235, 10, 71, 233, 10, 167, 82, 166, 22, 26, 80, 173, 73, 30, 222, 80, 106, 143, 127, 10, 80, 80, 199, 10, 164, 147, 113, 11, 37, 146, 120, 83, 151, 97, 252, 75, 80, 80, 131, 138, 236, 179, 77, 215, 250]

4. Візуалізувати масив за допомогою діаграми типу графік. Виділити кольором рівень, який відповідає значенню a, а також вказати значення на діаграмі. Виділити іншим кольором рівень, який відповідає значенню b, а також вказати значення на діаграмі.

```

In [27]: import matplotlib.pyplot as plt

def visualize_array(array, a, b):
# Створюємо графік
plt.figure(figsize=(10, 6))
x = range(len(array)) # Значення x - індекси елементів масиву

# Будуємо графік для всього масиву
plt.plot(x, array, label="Масив", color='blue', marker='o')

# Виділяємо рівень для a
a_indices = [i for i, value in enumerate(array) if value == a]
plt.scatter(a_indices, [a] * len(a_indices), color='red', label=f"Значення a")

# Виділяємо рівень для b
b_indices = [i for i, value in enumerate(array) if value == b]
plt.scatter(b_indices, [b] * len(b_indices), color='green', label=f"Значення b")

```

```

# Підписуємо осі
plt.xlabel("Індекс")
plt.ylabel("Значення")

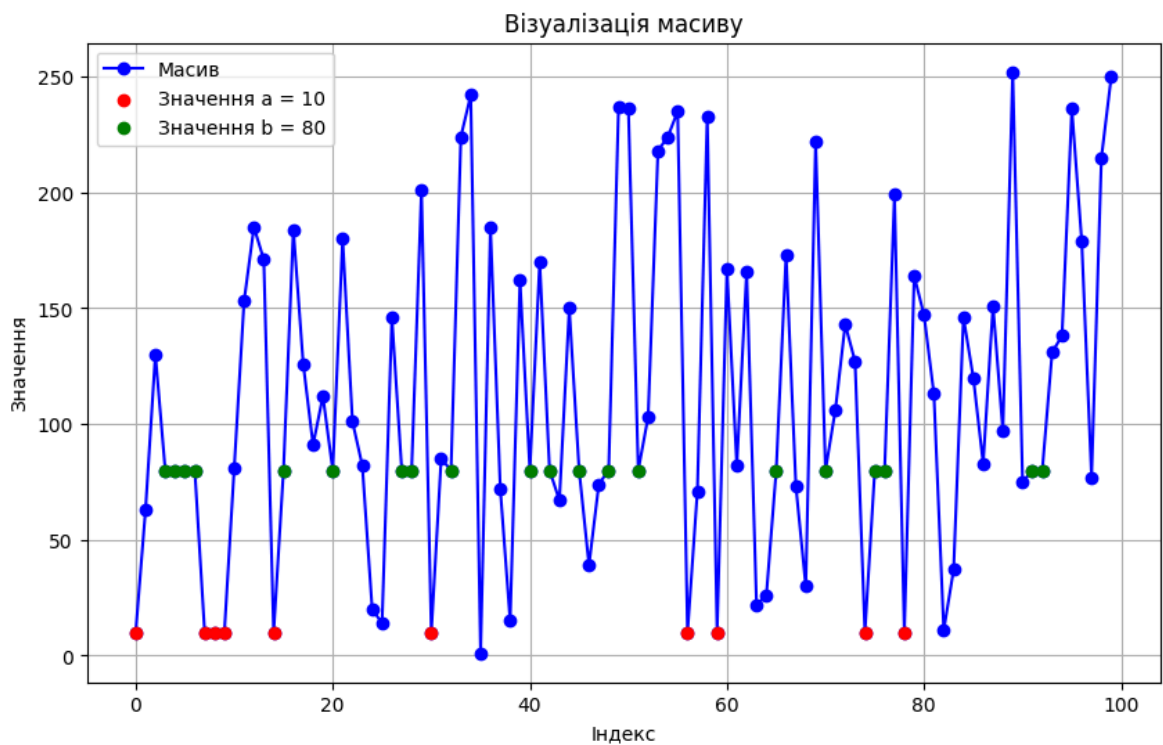
# Підписуємо діаграму
plt.title("Візуалізація масиву")

# Додаємо легенду
plt.legend()

# Показуємо графік
plt.grid(True)
plt.show()

# Візуалізуємо масив
visualize_array(array_from_file, a, b)

```



5. Порахувати кількість та суму елементів масиву, розташованих між елементами a та b.

```

In [28]: def count_and_sum_between_a_b(array, a, b):
# Переконаємося, що a < b для коректної логіки
if a > b:
    a, b = b, a

# Знайдемо елементи між значеннями a і b
elements_between = [x for x in array if a < x < b]

# Кількість елементів
count = len(elements_between)

# Сума елементів
total_sum = sum(elements_between)

```

```

    return count, total_sum

# Підраховуємо кількість та суму елементів між a та b
count, total_sum = count_and_sum_between_a_b(array_from_file, a, b)

# Виведення результату
print(f"Кількість елементів між {a} і {b}: {count}")
print(f"Сума елементів між {a} і {b}: {total_sum}")

```

Кількість елементів між 10 і 80: 17

Сума елементів між 10 і 80: 786

6. Порахувати середнє значення елементів масиву, розташованих до елемента a, середнє значення елементів масиву, розташованих між a та b, середнє значення елементів масиву, розташованих після елемента b. Вказати відповідні рівні на діаграмі.

```

In [29]: import matplotlib.pyplot as plt

def calculate_means(array, a, b):
    if a > b:
        a, b = b, a

    # Елементи до a
    elements_before_a = [x for x in array if x < a]

    # Елементи між a і b
    elements_between_a_b = [x for x in array if a < x < b]

    # Елементи після b
    elements_after_b = [x for x in array if x > b]

    # Середні значення
    mean_before_a = sum(elements_before_a) / len(elements_before_a) if elements_
    mean_between_a_b = sum(elements_between_a_b) / len(elements_between_a_b) if
    mean_after_b = sum(elements_after_b) / len(elements_after_b) if elements_aft

    return mean_before_a, mean_between_a_b, mean_after_b

def visualize_means_with_levels(array, a, b, mean_before_a, mean_between_a_b, me
plt.figure(figsize=(10, 6))
x = range(len(array)) # Індеси елементів масиву

# Будуємо графік для всього масиву
plt.plot(x, array, label="Масив", color='blue', marker='o')

# Виділяємо рівень для a
a_indices = [i for i, value in enumerate(array) if value == a]
plt.scatter(a_indices, [a] * len(a_indices), color='red', label=f"Значення a

# Виділяємо рівень для b
b_indices = [i for i, value in enumerate(array) if value == b]
plt.scatter(b_indices, [b] * len(b_indices), color='green', label=f"Значення

# Додаємо горизонтальні лінії для середніх значень
if mean_before_a is not None:
    plt.axhline(y=mean_before_a, color='orange', linestyle='--', label=f"Сер

```

```

if mean_between_a_b is not None:
    plt.axhline(y=mean_between_a_b, color='purple', linestyle='--', label=f"Середнє між a і b")

if mean_after_b is not None:
    plt.axhline(y=mean_after_b, color='brown', linestyle='--', label=f"Середнє після b")

# Підписуємо осі
plt.xlabel("Індекс")
plt.ylabel("Значення")

# Підписуємо діаграму
plt.title("Візуалізація масиву з середніми значеннями")

# Додаємо легенду
plt.legend()

# Показуємо графік
plt.grid(True)
plt.show()

# Обчислюємо середні значення
mean_before_a, mean_between_a_b, mean_after_b = calculate_means(array_from_file, a, b)

# Виведення середніх значень
print(f"Середнє значення елементів до a = {mean_before_a:.2f}" if mean_before_a else "")
print(f"Середнє значення елементів між a та b = {mean_between_a_b:.2f}" if mean_between_a_b else "")
print(f"Середнє значення елементів після b = {mean_after_b:.2f}" if mean_after_b else "")

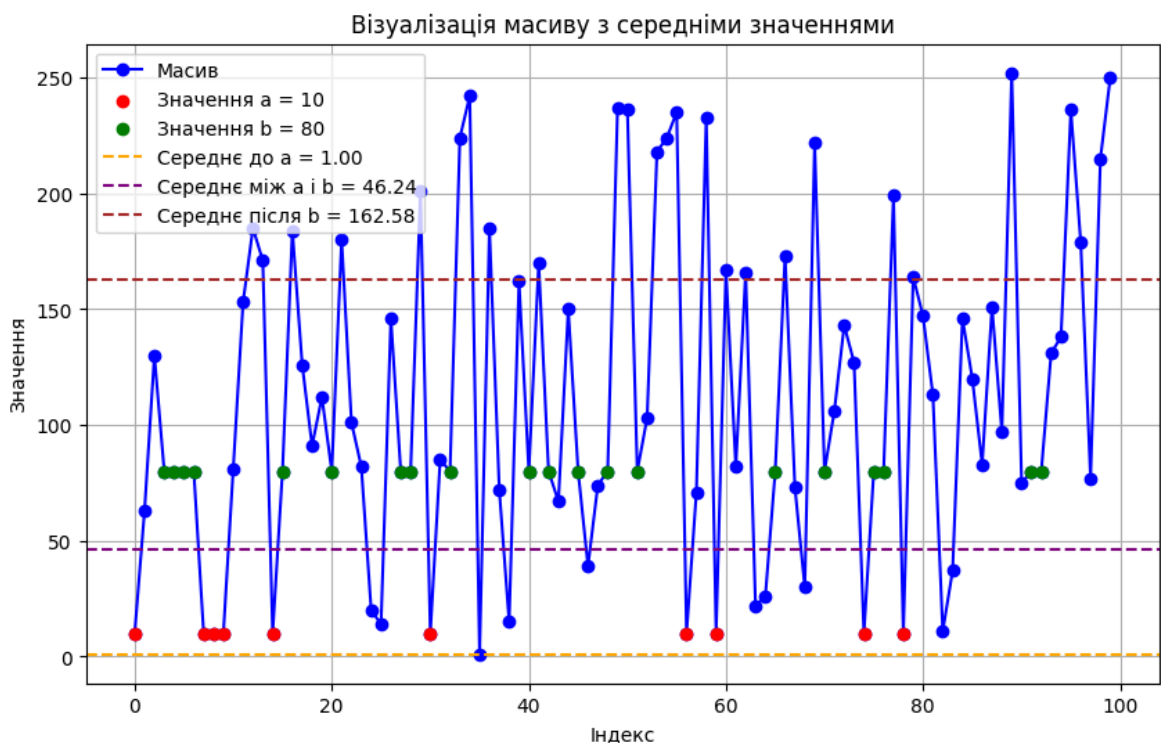
# Візуалізуємо масив з рівнями середніх значень
visualize_means_with_levels(array_from_file, a, b, mean_before_a, mean_between_a_b, mean_after_b)

```

Середнє значення елементів до a = 1.00

Середнє значення елементів між a та b = 46.24

Середнє значення елементів після b = 162.58



7. Побудувати діаграму яка містить мінімальний елемент, порашовані значення в пункті 6, значення a та b, максимальний елемент.

```
In [30]: import matplotlib.pyplot as plt

def build_summary_chart(min_value, mean_before_a, mean_between_a_b, mean_after_b, a, b, max_value):
    # Значення, які будемо відображати на діаграмі
    values = [min_value, mean_before_a, mean_between_a_b, mean_after_b, a, b, max_value]
    labels = ['Мінімум', 'Середнє до a', 'Середнє між a і b', 'Середнє після b', 'a', 'b', 'Максимум']

    # Фільтруємо None значення
    filtered_values = [v for v in values if v is not None]
    filtered_labels = [labels[i] for i in range(len(values)) if values[i] is not None]

    # Побудова горизонтальної діаграми
    plt.figure(figsize=(10, 6))
    plt.barh(filtered_labels, filtered_values, color=['blue', 'orange', 'purple', 'green', 'red', 'brown'])

    # Додаємо значення над кожною смугою
    for index, value in enumerate(filtered_values):
        plt.text(value, index, f'{value:.2f}', va='center', ha='right', color='black')

    # Налаштування осей
    plt.xlabel('Значення')
    plt.title('Ключові значення масиву')
    plt.grid(True)

    # Відображення діаграми
    plt.show()

# Обчислюємо середні значення
mean_before_a, mean_between_a_b, mean_after_b = calculate_means(array_from_file, 10, 100, 1000)

# Знаходимо мінімум та максимум
min_value = min(array_from_file)
max_value = max(array_from_file)

# Виводимо ключові значення
print(f"Мінімальний елемент: {min_value}")
print(f"Максимальний елемент: {max_value}")
print(f"Середнє значення елементів до a = {mean_before_a:.2f}" if mean_before_a else "Середнє значення елементів до a не обчислено")
print(f"Середнє значення елементів між a та b = {mean_between_a_b:.2f}" if mean_between_a_b else "Середнє значення елементів між a та b не обчислено")
print(f"Середнє значення елементів після b = {mean_after_b:.2f}" if mean_after_b else "Середнє значення елементів після b не обчислено")

# Побудова діаграми
build_summary_chart(min_value, mean_before_a, mean_between_a_b, mean_after_b, a, b, max_value)
```

Мінімальний елемент: 1

Максимальний елемент: 252

Середнє значення елементів до a = 1.00

Середнє значення елементів між a та b = 46.24

Середнє значення елементів після b = 162.58

