

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №6
із дисципліни «Основи програмування»
на тему
РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ

Виконав:
студент групи КМ-93
Пиндиківський Т. Р.

Керівник:
асистент Дрозденко О. М.

ЗМІСТ

МЕТА РОБОТИ.....	3
ПОСТАНОВКА ЗАДАЧІ.....	4
ОСНОВНА ЧАСТИНА	5
ВИСНОВКИ	6
ДОДАТОК А	7
ДОДАТОК Б	10

МЕТА РОБОТИ

Вивчити особливості роботи з одновимірними і двовимірними масивами із застосуванням рекурсивних функцій.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1:

Дано одновимірний масив числових значень, що нараховує n елементів. Виключити з масиву елементи, що належать проміжку $[b;c]$.

Завдання 2:

Виконати обробку елементів прямокутної матриці A , що має n рядків і m стовпців. Виключити з матриці рядок з номером k . Зімкнути рядки матриці.

ОСНОВНА ЧАСТИНА

Завдання 1 : Для виконання програми спочатку імпортуються модулі `re`, `array`, `random`, які слугують для обробки регулярних виразів, масивів та введення довільних значень чисел відповідно. Користувачу спочатку пропонується ввести розмірність масиву. Правильність введених даних перевіряється з використанням функції `validator`, яка обробляючи шаблон та отримане значення ввідних даних пропонує користувачу ввести правильні аргументи, поки не буде отримано підходящі значення. Потім відбувається заповнення масиву довільним цілими числами з проміжку `[0, 1000]`. Після чого користувач має вказати нижню та верхню межу інтервалу , елементи якого будуть виключені із початкового масиву. Ті елементи, які містяться у згаданому раніше інтервалі видаляються із масиву за допомогою функції `remove(element)`.

Завдання 2 : Для виконання програми спочатку імпортуються модулі `re`, `numpy`, які слугують для обробки регулярних виразів, масивів різної розмірності відповідно. Користувачу спочатку пропонується ввести кількість рядків та стовпців масиву. Правильність введених даних перевіряється з використанням функції `validator`, яка обробляючи шаблон та отримане значення ввідних даних пропонує користувачу ввести аргументи, поки не буде отримано підходящі значення. Потім відбувається заповнення двовимірного масиву довільними цілмими числами з проміжку `[0, 1000]`. Після чого користувач вводить номер рядку, який необхідно видалити, тоді і відбувається видалення рядку з масиву шляхом застосування функції `delete()`. В кінцевому результаті виводиться масив із зімкнутими рядками (відбувається застосування функції `flatten()`).

ВИСНОВКИ

На цій лабораторній роботі було вивчено особливості роботи з одновимірними і двовимірними масивами із застосуванням рекурсивних функцій. Були розроблені програми, які реалізують поставлену задачу відповідно до вимог. Їх можна застосовувати для вирішення однотипних завдань. Є багато можливостей для розширення функціоналу з метою забезпечення виконання складніших завдань.

ДОДАТОК А

1. Програмна реалізація задачі №1

```
import re
import array
import random

pattern=re.compile(r"[+-]?\d+$")

def init():
    print("\nЗавдання 1: Дано одновимірний масив числових значень, що
нараховує n елементів. "+
        "Виключити з масиву елементи, що належать проміжку
[b;c].")

def number_validator(prompt, pattern):
    number=int (validator(prompt, pattern))
    while number<=0:
        number=int (validator(prompt, pattern))
    return number

def validator(prompt, pattern):
    data=(input(prompt))
    while not bool (pattern.match(data)):
        data=(input(prompt))
    return data

def start1():
    init()
    number_elements=number_validator("\nВведіть розмірність
початкового масиву : ", pattern)

    numbers=array.array('q', [random.randrange(0, 1001, 1) for i in
range(number_elements)])
    final_numbers=array.array('q',[i for i in numbers])

    print("\nПочатковий масив з числами : ")
    print(list(numbers))

    start_point=number_validator("\nВведіть мінімальне значення
проміжку : ", pattern)
    finish_point=number_validator("\nВведіть максимальне значення
проміжку : ", pattern)

    for element in numbers:
        if (element>=start_point) and (element<=finish_point):
            final_numbers.remove(element)

    print("\nФінальний масив з числами, які не входять у проміжок [ "
+ str(start_point) + " ; " + str(finish_point) + " ]")
    print(list(final_numbers))
```

```

D:\Taras Pyndykivskiy\kpi\programming\programming_1\laboratory_works\#6
λ python menu.py

Лабораторна робота №6
Пиндиківський Тарас, КМ-93
16 варіант

Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою

Введіть номер програми задачі для запуску її виконання (1/2) : 1

Завдання 1: Дано одновимірний масив числових значень, що нараховує n елементів. Виключити з масиву елементи, що належать проміжку [b;c].

Введіть розмірність початкового масиву : -3 (pattern)
Введіть розмірність початкового масиву : 0.3 (pattern)
Введіть розмірність початкового масиву : 10

Початковий масив з числами :
[176, 357, 907, 507, 612, 738, 528, 569, 285, 574]

Введіть мінімальне значення проміжку : -9
Введіть мінімальне значення проміжку : 0.3
Введіть мінімальне значення проміжку : 200
Введіть максимальне значення проміжку : 800

Фінальний масив з числами, які не входять у проміжок [ 200 ; 800 ]
[176, 907]

Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою

```

Активация Windows

Рис. 1 - програмна реалізація задачі №1

2. Програмна реалізація задачі №2

```

"""
Виконати обробку елементів прямокутної матриці A, що має n рядків і m
стовпців. Виключити з матриці рядок з номером k. Зімкнути рядки
матриці.
"""
import re
import numpy

pattern=re.compile(r"[+]?\d+$")

def validator(pattern, prompt):
    data=input(prompt)
    while not bool(pattern.match(data)):
        data=input(prompt)
    return int(data)

def number_validator(pattern, prompt):
    number=validator(pattern, prompt)
    while (number<=0):
        number=validator(pattern, prompt)
    return number

def row_validator(pattern, prompt, number_rows):
    number=validator(pattern, prompt)
    while((number<0) or (number>number_rows-1)):
        number=validator(pattern, prompt)
    return number

```



```

def init():
    print("\nЗавдання 2: Виконати обробку елементів прямокутної
матриці A, що має n рядків і m "+
        "стовпців. Виключити з матриці рядок з номером k. Зімкнути
рядки матриці.")

def start2():
    init()
    number_rows=number_validator(pattern, "\nВведіть кількість рядків
масиву : ")
    number_columns=number_validator(pattern, "\nВведіть кількість
стовпців масиву : ")

    numbers=numpy.random.randint(0,1001,(number_rows, number_columns))
    print("\nПочатковий вигляд масиву з " + str(number_rows) + "
рядками та " + str(number_columns) + " стовпцями : ")
    print(numbers)

    row_to_eliminate=row_validator(pattern, "\nВведіть рядок, який
потрібно видалити : ", number_rows)

    numbers=numpy.delete(numbers, row_to_eliminate, 0)
    print("\nМасив після видалення " + str(row_to_eliminate) + " рядка
: ")
    print(numbers)

    numbers=numbers.flatten('C')
    print("\nВигляд масиву із зімкнутими рядками : ")
    print(numbers)

```

```

Лабораторна робота №6
Пиндиківський Тарас, КМ-93
16 варіант

Натисніть Enter, щоб продовжити, будь-що інше, щоб завершити роботу з програмою

Введіть номер програми задачі для запуску її виконання (1/2) : 2

Завдання 2: Виконати обробку елементів прямокутної матриці A, що має n рядків і m стовпців. Виключити з матриці рядок з номером k. Зімкнути рядки матриці.

Введіть кількість рядків масиву : 7

Введіть кількість стовпців масиву : 7

Початковий вигляд масиву з 7 рядками та 7 стовпцями :
[[ 80 847 958 159 518 557 487]
 [209 964 332 172 725 139 197]
 [734 456 656 739 363 762 487]
 [361 210 523 925 795 952 432]
 [182 472 123 390 792  42 162]
 [254 344 225 846 960 463  18]
 [724 405 299 236 814 145  22]]

Введіть рядок, який потрібно видалити : 5

Масив після видалення 5 рядка :
[[ 80 847 958 159 518 557 487]
 [209 964 332 172 725 139 197]
 [734 456 656 739 363 762 487]
 [361 210 523 925 795 952 432]
 [182 472 123 390 792  42 162]
 [724 405 299 236 814 145  22]]

Вигляд масиву із зімкнутими рядками :
[ 80 847 958 159 518 557 487 209 964 332 172 725 139 197 734 456 656 739
 363 762 487 361 210 523 925 795 952 432 182 472 123 390 792  42 162 724
 405 299 236 814 145  22]

```

Активация Windows
Перейдіть до розділу "Настройки", щоб активувати Windows

Рис. 2 - тестування задачі №2

ДОДАТОК Б

Питання для самоперевірки

1. Як створюється одновимірний масив? Як отримати доступ до будь-якого елементу масиву?

Одновимірний масив створюється командою `a=array.array('i',[1,2,3,4,5])`, де вказується тип елементів масиву та об'єкт, на основі елементів якого створюється масив.

2. Як створюється матриця? Як отримати доступ до будь-якого елементу матриці?

У NumPy існує багато способів створити масив. Один із найбільш простих - створити масив із звичайних списків або кортежів Python, використовуючи функцію `numpy.array()`. Функція `array()` трансформує вкладені послідовності в багатовимірні масиви.

3. Які існують модулі для роботи з масивами?

Для роботи з масивами існують модулі `array` та `numpy`.

4. Для чого можна використовувати генератори списків в масивах?

Генератори списків у масивах використовуються для заповнення масиву елементами, значення яких можна визначити заздалегідь.

5. Методи додавання елементів в масив.

Для додавання елементів в масив використовується метод `array.append(x)`

6. Що таке осі масиву?

«Осі» масиву – кількість вимірів конкретного масиву.

7. Як можна створити послідовність чисел в масиві?

Для створення послідовностей чисел, в NumPy є функція `arange()`, аналогічна вбудованій в Python `range()`, тільки замість списків вона повертає масиви, і приймає не тільки цілі значення. Взагалі, при використанні `arange()` з аргументами типу `float`, складно бути впевненим в тому, скільки елементів буде отримано (через обмеження точності чисел з плаваючою комою).

8. Яка функція створює одиничну матрицю?

Для створення одиничної двовимірної матриці використовується функція `eye()`.

9. Як можна змінити форму матриці?

Форму матриці можна змінити методом `reshape (rows, cols)`.

10. Як можна виконати об'єднання і розбиття матриці?

Для об'єднання матриці використовуються функції `hstack()` та `vstack()`. `hstack()` об'єднує масиви по горизонталі, `vstack()` - по вертикалі.

Використовуючи `hsplit()`, можна розбити масив вздовж горизонтальної осі, вказавши або кількість масивів, що повертаються однакової форми, або номера стовпців. Функція `vsplit()` розбиває масив вздовж вертикальної осі, а `array_split()` дозволяє вказати осі, уздовж яких відбудеться розбиття.

11. Як можна створити масив з випадкових елементів?

Найпростіший спосіб задати масив з випадковими елементами – використовувати функцію `sample` (або `random`, або `random_sample`, або `randf` - це одна і та ж функція).