

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

ЗВІТ

про виконання індивідуального завдання

з дисципліни "Навчальна практика"

студентки І-го курсу

групи ПМП-12

Трей Аліси

Керівник роботи

ас. Борисюк Я.Є.

Львів – 2019

Завдання 1.

Розв'язати квадратне рівняння виду $ax^2+bx+c=0$. Вивести розв'язки(дійсні корені) або повідомлення, що дійсних розв'язків немає(вивести комплексні числа)

Користуємося бібліотекою математичних функцій `import math`

Алгоритм:

Вводимо задані числа. Задаємо дві умови коли дискримінант більше 0 і дорівнює йому. Ще одна умова - коли дискримінант від'ємний. Виводимо корені

Програма:

```
import math
print("Введіть коефіцієнти квадратного рівняння: ")
a=float(input("a="))
b=float(input("b="))
c=float(input("c="))
d=b**2-4*a*c
if d>0:
    x1=(-b+math.sqrt(d))/(2*a)
    x2=(-b-math.sqrt(d))/(2*a)
    print('x1=', x1)
    print('x2=', x2)
elif d==0:
    x=-b/(2*a)
    print(x)
else:
    print("Комплексні корені: ")
    ai1=(math.sqrt((-1)*d))/(2*a)
    bi1=(-1*b)/(2*a)
    ai2=(-1)*math.sqrt((-1)*d)/(2*a)
    bi2=(-1*b)/(2*a)
    print('x1=',ai1,'i+',bi1)
    print('x2=',ai2,'i+',bi2)
```

Приклад:

Введіть коефіцієнти квадратного рівняння:

a= 2

b= 3

c= 4

Комплексні корені:

x1= 1.1989578808281798 i+ -0.75

x2= -1.1989578808281798 i+ -0.75

Завдання 2.(1)

При заданому значенні n та x знайти S . Тобто знайти суму перших n доданків ряду

$$S=1+2*3*4*5*x-3*4*5*6*x^2+\dots$$

Алгоритм:

Вводимо значення X та кількість елементів. Знаходимо через попередні значення елементи заданого прикладу. Виводимо сумму у вигляді
номер : значення

Програма:

```
x=float(input('X='))
n=int(input('N='))
d=1
s=1
a=1
b=-1
for i in range(2, n+1):
    a=i*(i+1)*(i+2)*(i+3)
    b=b*x*(-1)
    d=a*b
    s=s+d
    print(i,":",s)
print('S=',s)
```

Приклад:

```
X=2
N=5
2 : 241.0
3 : -1199.0
4 : 5521.0
5 : -21359.0
S=-21359.0
```

Завдання 2.(2)

При заданому значенні n та x знайти S . Тобто знайти суму перших n доданків ряду

$$S=1-(3/2!)*x^2+(9/4!)*x^4\dots$$

Використання своєї функції.

Інтерпретатор запускає деякий модуль (вихідний файл) як основну програму, він привласнює спеціальної змінної `__name__` значення `"__main__"`. Якщо цей файл імпортується з іншого модуля, змінної `__name__` буде присвоєно ім'я цього модуля.

Алгоритм:

Робим функцію з умовою факторіала. Вводимо дані X та N. Знаходимо через попередні значення елементи заданого прикладу. Виводимо сумму у вигляді
номер : значення

Програма:

```
def factorial(n):
    i=1
    r=1
    while(i<=n):
        r*=i
        i+=1
    return r
if __name__=="__main__":
    x=int(input("x="))
    n1=int(input("n="))
    s1=1
    s=1
    for i in range(n1+1):
        if i==0:
            print(i,":",s1)
        else:
            s1=(pow(-1,i))*(pow(x,(2*i)))*((2*(pow(i,2)))+1)/(factorial(2*i))
            s+=s1
            print(i,":",s1)
    print("S=",s)
```

Приклад:

```
x=2
n=5
0 : 1
1 : -6.0
2 : 6.0
3 : -1.6888888888888889
4 : 0.20952380952380953
5 : -0.014391534391534391
S= -0.4937566137566137
```

Завдання 3(1):

Знайти перших k натуральних чисел, більших від заданого числа A , в яких парні та непарні цифри чергуються. 0 - парна цифра

Використовуємо бібліотеку математичних функцій та створюємо власні.

Алгоритм:

Робимо функцію для простих чисел, потім для непарних, для парних та тих, що повторюються. Далі іде сама програма, де ми вводимо число, за яке всі подальші мають бути більшими, та їх кількість.

Програма:

```
import math
def prime(number):
    c=0
    for j in range(1, number+1):
        if number%j==0:
            c=c+1
    if c==2:
        s=True
    else:
        s=False
    return s
def odd(number):
    if number%2==1:
        s=True
    else:
        s=False
    return s
def even(number):
    if number%2==0:
        s=True
    else:
        s=False
    return s
def repeat(number):
    fact=True
    counter_even=0
    counter_odd=0
    memo=number
    list_number=[]
    for j in range(len(str(number))):
        list_number.append(number%10)
```

```

    number=int(number/10)
list_number.reverse()
number=memo
if even(list_number[0]):
    for j in range(2, len(str(number)), 2):
        if odd(list_number[j]):
            fact=False
            break
    for j in range(1, len(str(number)), 2):
        if even(list_number[j]):
            fact=False
            break
elif odd(list_number[0]):
    for j in range(1, len(str(number)), 2):
        if odd(list_number[j]):
            fact=False
            break
    for j in range(2, len(str(number)), 2):
        if even(list_number[j]):
            fact=False
            break
return fact
f=int(input('Мінімальний номер: ')) #вище задані функції, далі сама програма
k=int(input('Кількість чисел: '))
c=0
num=f
while c<k:
    if prime(num) and repeat(num):
        print(num)
        c=c+1
        num=num+1
    else:
        num=num+1

```

Приклад:

Мінімальний номер: 32

Кількість чисел: 6

41

43

47

61

67

83

Завдання 3(2):

Підрахувати кількість чисел Фібоначчі, що належать відріzkу $[a,b]$ і в записі яких є хоча б одна задана цифра K . Вивести дані числа

Використовуємо бібліотеку математичних функцій та створюємо власні для простих, непарних, парних та повторюючихся чисел. Створюємо умови для пошуку чисел Фібоначчі

Алгоритм:

Після заданих функцій вводимо значення числа, більше за яке будуть проводитись операції, значення числа, яке буде верхньою межею, та цифру, яке має бути хоча б один раз в записі числа

Програма:

```
import math
def prime(number):
    counter=0
    for j in range(1, number+1):
        if number%j==0:
            counter=counter+1
    if counter==2:
        state=True
    else:
        state=False
    return state
def odd(number):
    if number%2==1:
        state=True
    else:
        state=False
    return state
def even(number):
    if number%2==0:
        state=True
    else:
        state=False
    return state
def repeat(number):
    fact=True
    counter_even=0
    counter_odd=0
    memo=number
    list_number=[]
```

```

for j in range(len(str(number))):
    list_number.append(number%10)
    number=int(number/10)
list_number.reverse()
number=memo
if even(list_number[0]):
    for j in range(2, len(str(number)), 2):
        if odd(list_number[j]):
            fact=False
            break
    for j in range(1, len(str(number)), 2):
        if even(list_number[j]):
            fact=False
            break
elif odd(list_number[0]):
    for j in range(1, len(str(number)), 2):
        if odd(list_number[j]):
            fact=False
            break
    for j in range(2, len(str(number)), 2):
        if even(list_number[j]):
            fact=False
            break
return fact
lower_limit=int(input('Мінімальне число: '))
upper_limit=int(input('Максимальне: '))
present_number=str(input('Шукане число: '))
fibonacci=[]
fibonacci_pass=[]
string_check=""
fibonacci.append(0)
fibonacci.append(1)
count=0
count1=0
i=1
while fibonacci[i]<lower_limit:
    i=i+1
    fibonacci.append(fibonacci[i-1]+fibonacci[i-2])
memo1=i
while fibonacci[i]<upper_limit:
    i=i+1
    fibonacci.append(fibonacci[i-1]+fibonacci[i-2])
memo2=i
for k in range(memo1, memo2):
    fibonacci_pass.append(fibonacci[k])
    count1=count1+1

```



```
for k in range(count1):  
    if present_number in str(fibo_pass[k]):  
        print(fibo_pass[k])
```

Приклад:

Мінімальне число: 30
Максимальне: 1600
Шукане число: 7
377
987
1597

Завдання 4(1.1):

Дано стрічку, яка містить латинські слова, між словами - не менше одного пропуску(може бути довільна кількість пропусків), за останнім словом - крапка.
Вивести всі слова зі стрічки відсортовані в алфавітному порядку

Алгоритм:

Вводимо слова. За допомогою list ставимо пробіли. list.sort - сортує в алфавітному порядку.

Программа:

```
string=input("Введіть рядок: ")  
list1=string.split()  
print(list1)  
list1.sort()  
print(list1)
```

Приклад:

Введіть рядок: bear power apple sand game
['bear', 'power', 'apple', 'sand', 'game']
['apple', 'bear', 'game', 'power', 'sand']

Завдання 4(1.2):

Вивести всі слова зі стрічки відсортовані по довжині слова

Алгоритм:

Вводимо слова. Використовуємо key - це функція якій ми присвоюємо будь яку функцію(тут довжину). Сортуємо за заданою функцією

Програма:

```
string=input("Введіть рядок: ")
list1=string.split()
print(list1) list1.sort(key=len)
print(list1)
```

Приклад:

Введіть рядок: apple cow duck interesting clever
['apple', 'cow', 'duck', 'interesting', 'clever']
['cow', 'duck', 'apple', 'clever', 'interesting']

Завдання 4(2):

Створити список з тих слів стрічки, які відрізняються від останнього слова і виконується умова - слово симетричне. Вивести список. Видалити дані слова зі стрічки. Вивести оновлену стрічку

Алгоритм:

Робимо функцію для симетричних стрічок та для рівних стрічок.Видаляємо останнє слово, бо нам треба щоб всі інші були відмінні від нього, та дзеркальні . Повертаємо останнє за умовою

Програма:

```
def mirror(string):
    string_reverse=""
    for i in string[::-1]:
        string_reverse=string_reverse+i
    if string==string_reverse:
        state=True
    else:
        state=False
    return state
def not_equal(string1, string2):
    if string1==string2:
        state=False
    else:
        state=True
    return state
string=input('Введіть стрічку: ')
list_word=string.split()
print(list_word)
last_word=(list_word[len(list_word)-1]) #шукаємо останнє слово
```

```

list_pass=[]
list_new=[]
string_new=""
for i in range(len(list_word)-1):
    if mirror(list_word[i]) and not_equal(list_word[i], last_word):
        list_pass.append(list_word[i])
    else:
        list_new.append(list_word[i])
for i in range(len(list_new)):
    string_new=string_new+list_new[i]+' '
string_new=string_new+last_word    #вертаємо останнє слово за умовою
print(list_pass)
print(string_new)

```

Приклад:

Введіть стрічку: cool ola lol orange mum hello dog
 ['cool', 'ola', 'lol', 'orange', 'mum', 'hello', 'dog']
 ['lol', 'mum']
 cool ola orange hello dog

Завдання 5(1):

Задано масив цілих чисел А. Знайти серед елементів масиву А всі повні квадрати та утворити з них новий масив Б. Видалити з вихідного масиву А всі повні квадрати. Вивести вхідний масив А, отриманий масив Б та змінений масив А.

Використовуємо бібліотеку математичних функцій import math.

Алгоритм:

Робимо пустий список А. Додаємо елементи поки не буде елементу 0 (при вводі 0 програма перестає запрошувати числа). Видаляємо 0. Робимо пустий список Б та список індексів. Перевіряємо повні квадрати. Видаляємо з А. Ссовуємо індекси на 1 вліво. Виводимо списки А початкове, Б, А кінцеве

Програма:

```

import math
list_input=[]
t=1
while trade!=0:
    t=int(input("Введіть елементи: "))
    list_input.append(t)
list_input.pop(len(list_input)-1)
print('A: ')
print(list_input)

```

```

list_output=[]
list_index=[]
for i in range(len(list_input)):
    if math.sqrt(list_input[i])==int(math.sqrt(list_input[i])):
        list_output.append(list_input[i])
        list_index.append(i)
for i in range(len(list_index)):
    list_input.pop(list_index[i])
    for j in range(len(list_index)):
        list_index[j]=list_index[j]-1
print('Повні квадрати: ')
print(list_output)
print('Кінцевий A: ')
print(list_input)

```

Приклад:

A:
 [2, 6, 7, 8, 1, 16, 25, 30, 49, 9, 3, 17]
 Повні квадрати:
 [1, 16, 25, 49, 9]
 Кінцевий A:
 [2, 6, 7, 8, 30, 3, 17]

Завдання 5(2):

Задано стрічку, визначити скільки різних слів входить у стрічку(не враховувати регістр). Тобто, Student, student, sTUDENT, STUDENT - одне і теж слово. Вивести початкову стрічку (список всіх слів), кількість слів унікальних та список унікальних слів.

Алгоритм:

Вводимо стрічку. Записуємо елементи, які ми видалимо, якщо вони будуть в стрічці. Перевіряємо на наявність такого елемента стрічку. Записуємо її вже без них. Додаємо в словники кількість унікальних слів, унікальні слова. Виводимо початкову стрічку, унікальні слова, їх кількість.

Програма:

```

string_input=input('Введіть стрічку: ')
list_words=string_input.lower().split()
list_index=[]
new_list=[]
list_characters_new=[]
string_forbidden='!@#%&*(),.;:;?-_+= '
for i in range(len(list_words)):
    list_characters=list(list_words[i])

```

```

for j in range(len(list_characters)):
    if not(list_characters[j] in string_forbidden):
        list_characters_new.append(list_characters[j])
word="".join(char for char in list_characters_new)
new_list.append(word)
word=word[:0]
list_characters.clear()
list_characters_new.clear()
new_string=' '.join(word for word in new_list)
print(new_string)
word_dictionary={}
new_new_string=""
for i in range(len(new_list)):
    if not(new_list[i] in word_dictionary.keys()):
        word_dictionary[new_list[i]]=1
        new_new_string=new_new_string+new_list[i]+' '
    else:
        word_dictionary[new_list[i]]=word_dictionary[new_list[i]]+1
print('Кількість унікальних слів: ')
print(len(word_dictionary.keys()))
print('Унікальні слова: ')
print(new_new_string)

```

Приклад:

Введіть стрічку: hello Hello Hello! dog cat Dog salad cat@
hello hello hello dog cat dog salad cat
Кількість унікальних слів:
4
Унікальні слова:
hello dog cat salad

Завдання 5(3а):

Задано масив цілих чисел А. Знайти серед елементів масиву А всі числа Фібоначчі та утворити з них новий масив Б. Видалити з вихідного масиву А всі числа Фібоначчі. Вивести вхідний масив А, отриманий масив Б та змінений масив А.

Використовуємо бібліотеку математичних функцій import math.

Алгоритм:

Робимо пустий список А. Додаємо елементи поки не буде елементу 0 (при вводі 0 програма перестав запрошувати числа). Видаляємо 0. Робимо пустий список Б та список індексів. Перевіряємо чи є число числом Фібоначчі. Видаляємо з А. Ссовуємо індекси на 1 вліво. Виводимо списки А початкове, Б, А кінцеве.

Програма:

```
import math
list_input=[]
trade=1
while trade!=0:
    trade=int(input('Введіть елементи: '))
    list_input.append(trade)
list_input.pop(len(list_input)-1)
print()
print('Початковий список А')
print(list_input)
list_fibo=[0, 1]
list_output=[]
list_index=[]
while list_fibo[len(list_fibo)-1]<max(list_input):
    list_fibo.append(list_fibo[len(list_fibo)-1]+list_fibo[len(list_fibo)-2])
for i in range(len(list_input)):
    if list_input[i] in list_fibo:
        list_output.append(list_input[i])
        list_index.append(i)
for i in range(len(list_index)):
    list_input.pop(list_index[i])
    for j in range(len(list_index)):
        list_index[j]=list_index[j]-1
print('Вихідний список Б(числа фібоначі)')
print(list_output)
print('Кінцевий список А')
print(list_input)
```

Приклад:

Початковий список А

[1, 2, 3, 5, 8, 4, 6, 17, 34, 20]

Вихідний список Б(числа Фібоначчі)

[1, 2, 3, 5, 8, 34]

Кінцевий список А

[4, 6, 17, 20]

Завдання 5(3(б)):

Задано масив цілих чисел А. Знайти серед елементів масиву А всі числа Мерсенна та утворити з них новий масив Б. Видалити з вихідного масиву А всі числа Мерсенна. Вивести вхідний масив А, отриманий масив Б та змінений масив А.

Використовуємо бібліотеку математичних функцій import math.

Алгоритм:

Робимо пустий список А. Додаємо елементи поки не буде елементу 0 (при вводі 0 програма перестає запрошувати числа). Видаляємо 0. Робимо пустий список Б та список індексів. Перевіряємо чи є число числом Мерсенна. Видаляємо з А. Ссовуємо індекси на 1 вліво. Виводимо списки А початкове, Б, А кінцеве

Програма:

```
import math
list_input=[]
trade=1
while trade!=0:
    trade=int(input('Введіть елементи: '))
    list_input.append(trade)
list_input.pop(len(list_input)-1)
print()
print('Початковий список А')
print(list_input)
list_m=[1]
list_output=[]
list_index=[]
while list_m[len(list_m)-1]<max(list_input):
    list_m.append(pow(2,len(list_m)+1)-1)
for i in range(len(list_input)):
    if list_input[i] in list_m:
        list_output.append(list_input[i])
        list_index.append(i)
for i in range(len(list_index)):
    list_input.pop(list_index[i])
    for j in range(len(list_index)):
        list_index[j]=list_index[j]-1
print('Вихідний список Б(числа Мерсенна)')
print(list_output)
print('Кінцевий список А')
print(list_input)
```

Приклад:

Початковий список А
[1, 2, 3, 4, 5, 6, 7, 8, 9]
Вихідний список Б(числа Мерсенна)
[1, 3, 7]
Кінцевий список А
[2, 4, 5, 6, 8, 9]

Завдання 5(4):

Задано масив $L(n)$. Виключити з нього моду. Вивести вихідний масив та змінений.
Може бути не одна мода.

Використовуємо бібліотеку математичних функцій `import math`.

Алгоритм:

Робимо пустий список А. Додаємо елементи поки не буде елементу 0 (при вводі 0 програма перестає запрошувати числа). Видаляємо 0. Робимо пустий список Б та список індексів. Перевіряємо чи є число модою(зустрічається найчастіше в списку). Видаляємо з А. Ссовуємо індекси на 1 вліво. Виводимо списки А початкове, Б, А кінцеве.

Програма:

```
import math
list_input=[]
trade=1
while trade!=0:
    trade=int(input('Введіть елементи: '))
    list_input.append(trade)
list_input.pop(len(list_input)-1)
print()
print('Початковий список А')
print(list_input)
list_trend=[]
list_output=[]
list_index=[]
dictionary_numbers={}
dictionary_output={}
for i in range(len(list_input)):
    if not(list_input[i] in dictionary_numbers.keys()):
        dictionary_numbers[list_input[i]]=1
    else:
        dictionary_numbers[list_input[i]]=dictionary_numbers[list_input[i]]+1
trend_value=max(dictionary_numbers.values())
for key in dictionary_numbers.keys():
    if trend_value==dictionary_numbers.get(key):
        list_trend.append(key)
for i in range(len(list_input)):
    if list_input[i] in list_trend:
        list_output.append(list_input[i])
        list_index.append(i)
```



```

for i in range(len(list_index)):
    list_input.pop(list_index[i])
    for j in range(len(list_index)):
        list_index[j]=list_index[j]-1
for value in list_output:
    dictionary_output[value]='whatever'
print('Вихідний список Б(мода):')
print(list(dictionary_output.keys()))
print('Кінцевий список А:')
print(list_input)

```

Приклад:

Початковий список А:
 [5, 4, 6, 6, 1, 2, 3, 4, 3, 8, 3, 9, 6]
 Вихідний список Б(мода):
 [6, 3]
 Кінцевий список А:
 [5, 4, 1, 2, 4, 8, 9]

Завдання 6:

Написати словник “База логінів”. який відображає по введеному логіну пароль користувача, якщо такого слова(логіну) немає, то додаємо його у список.

Алгоритм:

Створюємо пустий словник. Відкриваємо текстовий файл. Назначаємо елементи меню. Кожному елементу присвоюємо значення та описуємо їх.

Програма:

```

Dict={}
with open('6(1).txt','r') as file: #r-читання w-запис a-дод в кінець
    for line in file:
        key,*value=line.split()
        Dict[key]=value
key=""
print('***20,' База логінів ', ***20)
print()
print('HELP:')
print()
print('Наберіть логін - щоб вивести пароль або команду у вигляді букви')
print()
print('"a" -щоб додати новий логін')

```

```

print()
print("'d" -щоб видалити логін - пароль')
print()
print("'l" -щоб переглянути увесь список')
print()
print("'b" -щоб переглянути список пар ,в яких логін починається з букви: ')
print()
print("'e" -щоб завершити роботу та записати словник у файл')
print()
while True:
    key=input("Введіть логін для виводу паролю: ")
    if key in Dict:
        print("Пароль: ", Dict.get(key))
    elif key=='a':
        newWord= input("Новий логін: ")
        Dict[newWord]=input("Пароль: ")
    elif key=='l':
        for item, trans in Dict.items():
            print(item, ': ', trans)
    elif key=='b':
        letter=input("Введіть букву на яку має починатись логін ")
        for i in Dict.keys():
            if i[0]==letter:
                print(i,":",Dict[i])
    elif key=='d':
        del Dict[input('Введіть логін або пароль для видалення: ')]
    elif key=='e':
        with open('6(1).txt', 'w') as out:
            for key,val in Dict.items():
                out.write('{}:{}'.format(key,val))
        print("Exit!")
        exit(0)
    else:
        print("Нема такого логіну. Додати?")
        key1=input("Введіть Y або N: ")
        if key1=='Y':
            Dict[key]=input("Пароль: ")
            print("Додано до словника", key, "-", Dict.get(key) )

```

Приклад:

***** База логінів *****

HELP:

Наберіть логін - щоб вивести пароль або команду у вигляді букви

"a" -щоб додати новий логін

"d" -щоб видалити логін - пароль

"l" -щоб переглянути увесь список

"b" -щоб переглянути список пар ,в яких логін починається з букви:

"e" -щоб завершити роботу та записати словник у файл

Введіть логін для виводу паролю: l

Nadiya123 : ['ert5677']

Cat234 : ['cat_cat12']

Введіть логін для виводу паролю: a

Новий логін: alisa

Пароль: trey

Введіть логін для виводу паролю: l

Nadiya123 : ['ert5677']

Cat234 : ['cat_cat12']

alisa : trey

Введіть логін для виводу паролю: b

Введіть букву на яку має починатись логін a

alisa : trey

Введіть логін для виводу паролю: d

Введіть логін або пароль для видалення: alisa

Введіть логін для виводу паролю: l

Nadiya123 : ['ert5677']

Cat234 : ['cat_cat12']

Завдання 7(2,3,4,5,7,8,9):

Алгоритм: створюємо функцію для виводу матриці, для генерації списку з колонок, для піднесення до 2 степеня матриці, для додавання двох матриць, для множення двох матриць, для множення матриці на 10, для транспонування матриць. Вводимо матриці А та В за початкових умов. Виводимо результати по черзі на екран.

Програма:

```
def print_matrix(matrix):  
    for i in range(len(matrix)):  
        for j in range(len(matrix[0])):
```

```

        print(matrix[i][j],end=' ')
    print()
    pass
def matrix_pow(matrix,k=2):
    result=[0]*len(matrix)
    for i in range(len(result)):
        result[i]=[0]*len(matrix[0])
    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            result[i][j]=matrix[i][j]
    for i in range(k-1):
        result=multiply_matrix(matrix,result)
    return result
def add_matrix(matrix1,matrix2):
    if (len(matrix1)!=len(matrix2))or(len(matrix1[0])!=len(matrix2[0])):
        print('Матриці не сумісні')
    else:
        result=[0]*len(matrix1)
        for i in range(len(result)):
            result[i]=[0]*len(matrix1[0])
        for i in range(len(matrix1)):
            for j in range(len(matrix1[0])):
                result[i][j]=matrix1[i][j]+matrix2[i][j]
    return result
def matrix_times_constant(matrix,k=10):
    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            matrix[i][j]=k*matrix[i][j]
    for i in range(len(matrix)):
        for j in range(len(matrix[i])):
            print(matrix[i][j],end=" ")
        print(" ")
    print()
    return matrix
def gen_list_columns(matrix):
    transposed=[0]*len(matrix[0])
    for i in range(len(transposed)):
        transposed[i]=[0]*len(matrix)
    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            transposed[j][i]=matrix[i][j]
    return transposed

```

```

def row_times_column(row,column):
    list_multiplicants=[]
    for k in range(len(row)):
        list_multiplicants.append(row[k]*column[k])
    result=sum(list_multiplicants)
    return result
def multiply_matrix(matrix1,matrix2):
    if len(matrix1[0])!=len(matrix2):
        print('Матриці не сумісні')
        result=0
    else:
        list_pairs=[]
        result=[0]*len(matrix1)
        for i in range(len(result)):
            result[i]=[0]*len(matrix2[0])
            for i in range(len(result)):
                for j in range(len(result[0])):
                    result[i][j]=row_times_column(matrix1[i],gen_list_columns(matrix2)[j])
                    list_pairs.clear()
        return result
A=[[1,2,3],[3,4,5],[6,7,8]]
B=[[1,2,8],[3,4,5],[6,7,8]]
A1=[[1,2,3],[3,4,5],[6,7,8]]
A2=[[1,2,3],[3,4,5],[6,7,8]]
A3=[[1,2,3],[3,4,5],[6,7,8]]
B3=[[1,2,8],[3,4,5],[6,7,3]]
print('A:')
for i in range(len(A)):
    for j in range(len(A[i])):
        print(A[i][j],end=" ")
    print(" ")
print()
print('B:')
for i in range(len(B)):
    for j in range(len(B[i])):
        print(B[i][j],end=" ")
    print(" ")
print()
print('Множення матриць A і B')
print(print_matrix(multiply_matrix(A,B)))
print("")
print('Множення матриці A на 10')

```

```

print(print_matrix(matrix_times_constant(A1)))
print("")
print('Підняття матриці A в 2 степінь')
print(print_matrix(matrix_pow(A2)))
print("")
print('Додавання матриць A і B')
print(print_matrix(add_matrix(A3,B3)))
print("")
def transpose(matr):
    res=[]
    n=len(matr)
    m=len(matr[0])
    for j in range(m):
        tmp=[]
        for i in range(n):
            tmp=tmp+[matr[i][j]]
        res=res+[tmp]
    return res
print('Транспонування матриці A')
print()
m = [[1,2,3],[3,4,5],[6,7,8]]
for i in range(len(m)):
    for j in range(len(m[i])):
        print(m[i][j],end=" ")
    print(" ")
print()
z=transpose(m)
for i in range(len(m[0])):
    for j in range(len(m)):
        print(z[i][j],end=' ')
    print()
print()

```

Приклад:

A:

1 2 3

3 4 5

6 7 8

B:

1 2 8

3 4 5

6 7 8

Множення матриць

25 31 42

45 57 84

75 96 147

None

Множення на 10

10 20 30

3 4 5

6 7 8

10 20 30

30 40 50

6 7 8

10 20 30

30 40 50

60 70 80

10 20 30

30 40 50

60 70 80

None

Підняття в 2 степінь

25 31 37

45 57 69

75 96 117

None

Додавання матриць

2 4 11

6 8 10

12 14 11

None

Транспонування матриці

1 2 3

3 4 5

6 7 8

1 3 6

2 4 7

3 5 8