**Національний технічний університет України**

**«Київський політехнічний інститут ім. Ігоря Сікорського»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №5**

*Транспонування та множення матриць*

*Виконав:Андрій Братун*

*Група:* ІО-63

**2016Теоретичні відомості**

***Введення та виведення матриць:***

***Матриця*** – двомірний масив, елементи котрого розташовуються в рядках та стовпцях, тобто кожний елемент має 2 координати. В дійсності ж не існує двомірних структур в пам’яті, масив записується лінійно, виділяючи лише 2 координати.

*Наприклад,* місця елементів в матриці 3Х3:

|  |  |  |  |
| --- | --- | --- | --- |
|  | a[1,1] | a[1,2] | a[1,3] |
| A = | a[2,1] | a[2,2] | a[2,3] |
|  | a[3,1] | a[3,2] | a[3,3] |

Кожний елемент має 2 координати – i (рядки) та j (стовпці).

*Наприклад*, елемент а[2,1] має координати i=2, j=1.

Елементами матриці можуть бути будь-які дійсні числа.

***Транспонування:***

***Транспонованою матрицею (АТ)*** називається матриця, що виникає з заданої матриці в результаті заміни її рядків на стовпчики та стовпчиків на рядки.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 |  | 1 | 4 | 7 |
| *Наприклад,* **A =** | 4 | 5 | 6 | **АТ** = | 2 | 5 | 8 |
|  | 7 | 8 | 9 |  | 3 | 6 | 9 |

Для транспонування матриці достатньо змінити координати елементів заданої матриці так, щоб координата стовпця стала координатою рядка, а координата рядка – координатою стовпця.

***Множення матриць:***

Нехай задано матриці А та В:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A =** | a[1,1] | a[1,2] | … | a[1,n] | **B =** | b[1,1] | b[1,2] | … | b[1,q] |
| a[2,1] | a[2,2] | … | a[2,n] | b[2,1] | b[2,2] | … | b[2,q] |
| … | … | … | … | … | … | … | … |
| a[m,1] | a[m,2] | … | a[m,n] | b[n,1] | b[n,2] | … | b[n,q] |

Тоді матриця С розміром mXq називається добутком матриць А та В:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| С = | с[1,1] | с[1,2] | … | с[1,q] |
| c[2,1] | c[2,2] | … | c[2,q] |
| … | … | … | … |
| c[m,1] | c[m,2] | … | c[m,q] |

, де С[i,j] = (i = 1, 2, …, m; j = 1, 2, …, q).

**Зауваження.** Операція множення справедлива лише тоді і тільки тоді, коли число стовпців в першому співмножнику дорівнює числу рядків в другому.

У випадку коли обидві матриці квадратні одного і того ж порядку, операція множення має місце.

Також існування добутку АВ не гарантує існування добутку ВА та АВ≠ВА.

Код програми:

**#Транспонування**

**import** random  
  
N = int(input(str('Input the number of lines -> ')))  
M = int(input(str('Input the number of column -> ')))  
matrix = [[random.randint(0, 10) **for** j **in** range(M)] **for** i **in** range(N)]  
  
print('Matrix -> {}'.format(matrix))  
  
  
**for** i **in** range(len(matrix)):  
 **for** j **in** range(0,i+1):  
 matrix[i][j], matrix[j][i] = matrix [j][i], matrix[i][j]  
  
print(matrix)

**#Множення**

**import** random  
  
N = int(input(str('Input the number of lines -> ')))  
M = int(input(str('Input the number of column -> ')))  
m1 = [[random.randint(0, 10) **for** j **in** range(M)] **for** i **in** range(N)]  
  
m2 = [[random.randint(0, 10) **for** k **in** range(M)] **for** m **in** range(N)]  
  
print(m1)  
print(m2)  
  
**def matrixmult**(m1,m2):  
 s = 0 # сумма  
 t = [] # временная матрица  
 m3 = [] # конечная матрица  
 **if** len(m2) != len(m1[0]):  
 print("Матрицы не могут быть перемножены")  
 **else**:  
 r1 = len(m1) # количество строк в первой матрице  
 c1 = len(m1[0]) # Количество столбцов в 1  
 r2 = c1 # и строк во 2ой матрице  
 c2 = len(m2[0]) # количество столбцов во 2ой матрице  
 **for** z **in** range(0, r1):  
 **for** j **in** range(0, c2):  
 **for** i **in** range(0, c1):  
 s = s+m1[z][i]\*m2[i][j]  
 t.append(s)  
 s = 0  
 m3.append(t)  
 t = []  
 **return** m3  
  
matrix3 = matrixmult(m1, m2)  
print(matrix3)

**Висновок:**

Оптимізацією даної програми є використання наборів функцій. Переваги використання функцій користувача очевидні: по-перше, щоб уникнути зайвих змінних і навіть операцій (звідки і слідує виграш в пам’яті), по-друге, для структурування програми, виділення окремих операцій і блоків.