

Матрице

Задатак 01: Написати програм који коришћењем функција креира квадратну матрицу реда n у којој су сви елементи на главној дијагонали једнаки 1, изнад ње једнаки 2 а испод ње једнаки 3.

```
def unos_dimenzije_matrice():
    return int(input("Uneti dimenziju kvadratne matrice: "))

def inicijalizacija_matrice(x):
    M = [[0 for i in range(x)] for j in range(x)]
    return M

def popunjavanje_matrice(x, M):
    for i in range(x):
        for j in range(x):
            if i == j:
                M[i][j] = 1
            else:
                if i < j:
                    M[i][j] = 2
                else:
                    M[i][j] = 3
    return M

def prikaz_rezultata(M):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

def main():
    n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(n)
    A = popunjavanje_matrice(n, A)
    prikaz_rezultata(A)
```

```
main()
```

```
Uneti dimenziju kvadratne matrice: 4
```

```
1      2      2      2
3      1      2      2
3      3      1      2
3      3      3      1
```

Задатак 02: Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима а затим и приказује матрицу.

```
def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija_matrice(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    return M

def popunjavanje_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
```

```

        M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def prikaz_rezultata(M):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

def main():
    m, n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(m, n)
    A = popunjavanje_matrice(m, n, A)
    prikaz_rezultata(A)

```

```
main()
```

Uneti broj redova matrice: 3

Uneti broj kolona matrice: 4

A[0][0] = 1

A[0][1] = 2

A[0][2] = 3

A[0][3] = 4

A[1][0] = 5

A[1][1] = 6

A[1][2] = 7

A[1][3] = 8

A[2][0] = 9

A[2][1] = 10

A[2][2] = 11

A[2][3] = 12

1 2 3 4

5 6 7 8

9 10 11 12

Задаци за самосталан рад:

1) Написати програм који коришћењем функција креира квадратну матрицу реда 5, попуњава је са 2 у сваку другу колону, са 1 у места помоћне дијагонале која нису део сваке друге колоне и са 0 преостала места у матрици.

2) Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима тако што ако је збир индекса позиције елемента између 5 и 8 убацује 1 а иначе се убацује 0, и приказује матрицу.

Задатак 03: Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима, израчунава суму сваког реда и приказује матрицу и израчунате суме.

```

def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    N = [0 for i in range(x)]

```

```

    return M, N

def popunjavanje_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def racunanje_sume_redova(x, y, M, N):
    for i in range(x):
        for j in range(y):
            N[i] += M[i][j]

    return N

def prikaz_rezultata(M, N):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print(N)

def main():
    m, n = unos_dimenzije_matrice()
    A, S = inicijalizacija(m, n)
    A = popunjavanje_matrice(m, n, A)
    S = racunanje_sume_redova(m, n, A, S)
    prikaz_rezultata(A, S)

```

```
main()
```

```
Uneti broj redova matrice: 3
```

```
Uneti broj kolona matrice: 4
```

```
A[0][0] = 1
```

```
A[0][1] = 2
```

```
A[0][2] = 3
```

```
A[0][3] = 4
```

```
A[1][0] = 5
```

```
A[1][1] = 6
```

```
A[1][2] = 7
```

```
A[1][3] = 8
```

```
A[2][0] = 9
```

```
A[2][1] = 10
```

```
A[2][2] = 11
```

```
A[2][3] = 12
```

```
1      2      3      4
```

```
5      6      7      8
```

```
9      10     11     12
```

```
[10, 26, 42]
```

Задатак 04: Написати програм за израчунавање сума елемената матрице који се налазе изнад и посебне суме елемената који се налазе испод помоћне дијагонале квадратне матрице реда n .

```
def unos_dimenzije_matrice():
    return int(input("Uneti dimenziju kvadratne matrice: "))

def inicijalizacija_matrice(x):
    M = [[0 for i in range(x)] for j in range(x)]
    return M

def popunjavanje_matrice(x, M):
    for i in range(x):
        for j in range(x):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def racunanje_sume_iznad(x, M):
    suma = 0
    for i in range(x):
        for j in range(x):
            if i + j < x - 1:
                suma += M[i][j]
    return suma

def racunanje_sume_ispod(x, M):
    suma = 0
    for i in range(x):
        for j in range(x):
            if i + j > x - 1:
                suma += M[i][j]
    return suma

def prikaz_rezultata(M, suma1, suma2):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print("Zbir elemenata iznad pomocne dijagonale je", suma1)
    print("Zbir elemenata ispod pomocne dijagonale je", suma2)

def main():
    n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(n)
    A = popunjavanje_matrice(n, A)
    suma_iznad = racunanje_sume_iznad(n, A)
    suma_ispod = racunanje_sume_ispod(n, A)
    prikaz_rezultata(A, suma_iznad, suma_ispod)

main()
```

Uneti dimenziju kvadratne matrice: 3

$A[0][0] = 1$

$A[0][1] = 2$

$A[0][2] = 3$

$A[1][0] = 1$

$A[1][1] = 2$

$A[1][2] = 3$

$A[2][0] = 1$

$A[2][1] = 2$

$A[2][2] = 3$

1 2 3

1 2 3

1 2 3

Zbir elemenata iznad pomocne dijagonale je 4

Zbir elemenata ispod pomocne dijagonale je 8

Задаци за самосталан рад:

3) Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима, израчунава суму сваке колоне и приказује матрицу и израчунате суме.

4) Написати програм који коришћењем функција креира квадратну матрицу реда n , попуњава је елементима, израчунава суму главне и помоћне дијагонале и приказује матрицу и израчунате суме.

Задатак 05: Написати програм који после уноса матрице реда $m \times n$ креира нову матрицу чији су елементи квадрати елемената прве матрице.

```
def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija_matrica(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    N = [[0 for j in range(y)] for i in range(x)]
    return M, N

def popunjavanje_osnovne_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def popunjavanje_izvedene_matrice(x, y, M, N):
    for i in range(x):
        for j in range(y):
            N[i][j] = M[i][j] ** 2

    return N

def prikaz_rezultata(M, N):
```

```

print("Osnovna matrica: ")
for red in M:
    for kolona in red:
        print(str(kolona), "\t", end = " ")
    print()
print()
print("Izvedena matrica: ")
for red in N:
    for kolona in red:
        print(str(kolona), "\t", end = " ")
    print()

def main():
    m, n = unos_dimenzije_matrice()
    A, B = inicijalizacija_matrice(m, n)
    A = popunjavanje_osnovne_matrice(m, n, A)
    B = popunjavanje_izvedene_matrice(m, n, A, B)
    prikaz_rezultata(A, B)

```

```

main()
Uneti broj redova matrice: 2
Uneti broj kolona matrice: 3
A[0][0] = 1
A[0][1] = 1
A[0][2] = 1
A[1][0] = 2
A[1][1] = 2
A[1][2] = 2
Osnovna matrica:
1      1      1
2      2      2

Izvedena matrica:
1      1      1
4      4      4

```

Задатак 06: Написати програм који открива колико је елемената матрице парно, непарно, једнако 0, веће од 0 и мање од 0.

```

def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija_matrice(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    return M

def popunjavanje_matrice(x, y, M):

```

```

for i in range(x):
    for j in range(y):
        M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

return M

def prebrojavanje_po_parnosti(x, y, M):
    par, npar, nul = 0, 0, 0
    for i in range(x):
        for j in range(y):
            if M[i][j] == 0:
                nul += 1
            elif M[i][j] % 2 == 0:
                par += 1
            elif M[i][j] % 2 == 1:
                npar += 1

    return par, npar, nul

def prebrojavanje_po_vrednosti(x, y, M):
    veci, manji = 0, 0
    for i in range(x):
        for j in range(y):
            if M[i][j] > 0:
                veci += 1
            elif M[i][j] < 0:
                manji += 1

    return veci, manji

def prikaz_rezultata(M, par, npar, nul, veci, manji):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print("Broj parnih elemenata u matrici je", par)
    print("Broj neparnih elemenata u matrici je", npar)
    print("Broj nula elemenata u matrici je", nul)
    print("Broj elemenata vecih od nula u matrici je", veci)
    print("Broj elemenata manjih od nula u matrici je", manji)

def main():
    m, n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(m, n)
    A = popunjavanje_matrice(m, n, A)
    parni, neparni, nule = prebrojavanje_po_parnosti(m, n, A)
    veci_od_0, manji_od_0 = prebrojavanje_po_vrednosti(m, n, A)
    prikaz_rezultata(A, parni, neparni, nule, veci_od_0, manji_od_0)

main()

```

Uneti broj redova matrice: 2

Uneti broj kolona matrice: 3

$A[0][0] = -2$

$A[0][1] = 10$

$A[0][2] = 0$

$A[1][0] = 3$

$A[1][1] = 5$

$A[1][2] = 0$

-2 10 0

3 5 0

Broj parnih elemenata u matrici je 2

Broj neparnih elemenata u matrici je 2

Broj nula elemenata u matrici je 2

Broj elemenata vecih od nula u matrici je 3

Broj elemenata manjih od nula u matrici je 1

Задаци за самосталан рад:

5) Написати програм који од матрице реда $m \times n$ одсецањем креира квадратну матрицу датог реда p ($p \leq \min(m,n)$).

6) Написати програм који коришћењем функција смештених у модуле попуњава матрицу димензије $m \times n$, приказује матрицу као и суме по колонама и редовима матрице.

Задаци за самосталан рад:

1) Написати програм који коришћењем функција креира квадратну матрицу реда 5, попуњава је са 2 у сваку другу колону, са 1 у места помоћне дијагонале која нису део сваке друге колоне и са 0 преостала места у матрици.

```
def inicijalizacija_matrice():
    M = [[0 for i in range(5)] for j in range(5)]
    return M

def popunjavanje_matrice(M):
    for i in range(5):
        for j in range(5):
            if j % 2 == 1:
                M[i][j] = 2
            elif i + j == 4:
                M[i][j] = 1

    return M

def prikaz_rezultata(M):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

def main():
    A = inicijalizacija_matrice()
    A = popunjavanje_matrice(A)
    prikaz_rezultata(A)
```

main()

```
0      2      0      2      1
0      2      0      2      0
0      2      1      2      0
0      2      0      2      0
1      2      0      2      0
```

2) Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима тако што ако је збир индекса позиције елемента између 5 и 8 убацује 1 а иначе се убацује 0, и приказује матрицу.

```
def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija_matrice(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    return M

def popunjavanje_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
            if (i + j >= 5) and (i + j <= 8):
                M[i][j] = 1

    return M

def prikaz_rezultata(M):
    for red in M:
```

```

    for kolona in red:
        print(str(kolona), "\t", end = " ")
    print()

```

```

def main():
    m, n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(m, n)
    A = popunjavanje_matrice(m, n, A)
    prikaz_rezultata(A)

```

```
main()
```

```

Uneti broj redova matrice: 4
Uneti broj kolona matrice: 5
0      0      0      0      0
0      0      0      0      1
0      0      0      1      1
0      0      1      1      1

```

3) Написати програм који коришћењем функција креира матрицу реда $m \times n$, попуњава је елементима, израчунава суму сваке колоне и приказује матрицу и израчунате суме.

```

def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def inicijalizacija(x, y):
    M = [[0 for j in range(y)] for i in range(x)]
    N = [0 for i in range(y)]
    return M, N

def popunjavanje_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def racunanje_sume_kolona(x, y, M, N):
    for i in range(x):
        for j in range(y):
            N[j] += M[i][j]

    return N

def prikaz_rezultata(M, N):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print(N)

def main():
    m, n = unos_dimenzije_matrice()
    A, S = inicijalizacija(m, n)

```

```
A = popunjavanje_matrice(m, n, A)
S = racunanje_sume_kolona(m, n, A, S)
prikaz_rezultata(A, S)
```

```
main()
Uneti broj redova matrice: 4
Uneti broj kolona matrice: 5
A[0][0] = 0
A[0][1] = 1
A[0][2] = 0
A[0][3] = 1
A[0][4] = 0
A[1][0] = 1
A[1][1] = 0
A[1][2] = 1
A[1][3] = 0
A[1][4] = 1
A[2][0] = 0
A[2][1] = 1
A[2][2] = 0
A[2][3] = 1
A[2][4] = 0
A[3][0] = 1
A[3][1] = 0
A[3][2] = 1
A[3][3] = 0
A[3][4] = 1
0      1      0      1      0
1      0      1      0      1
0      1      0      1      0
1      0      1      0      1
[2, 2, 2, 2, 2]
```

4) Написати програм који коришћењем функција креира квадратну матрицу реда n, попуњава је елементима, израчунава суму главне и помоћне дијагонала и приказује матрицу и израчунате суме.

```
def unos_dimenzije_matrice():
    return int(input("Uneti dimenziju kvadratne matrice: "))

def inicijalizacija_matrice(x):
    M = [[0 for i in range(x)] for j in range(x)]
    return M

def popunjavanje_matrice(x, M):
    for i in range(x):
        for j in range(x):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def racunanje_suma_po_dijagonalama(x, M):
    glav, pom = 0, 0
    for i in range(x):
        for j in range(x):
```

```

        if i == j:
            glav += M[i][j]
        if i + j == x - 1:
            pom += M[i][j]

    return glav, pom

def prikaz_rezultata(M, glav, pom):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print("Zbir elemenata glavne dijagonale je", glav)
    print("Zbir elemenata pomocne dijagonale je", pom)

def main():
    n = unos_dimenzije_matrice()
    A = inicijalizacija_matrice(n)
    A = popunjavanje_matrice(n, A)
    glavna, pomocna = racunanje_suma_po_dijagonalama(n, A)
    prikaz_rezultata(A, glavna, pomocna)

main()

```

Uneti dimenziju kvadratne matrice: 3

A[0][0] = 2

A[0][1] = -1

A[0][2] = 0

A[1][0] = 1

A[1][1] = 1

A[1][2] = 1

A[2][0] = 2

A[2][1] = 0

A[2][2] = 2

2 -1 0

1 1 1

2 0 2

Zbir elemenata glavne dijagonale je 5

Zbir elemenata pomocne dijagonale je 3

5) Написати програм који од матрице реда $m \times n$ одсецањем креира квадратну матрицу датог реда p ($p \leq \min(m, n)$).

```

def unos_dimenzije_matrice():
    x = int(input("Uneti broj redova matrice: "))
    y = int(input("Uneti broj kolona matrice: "))
    return x, y

def unos_dimenzije_kvadratne_matrice(x, y):
    while True:
        kvad = int(input("Uneti dimenziju kvadratne matrice: "))
        if kvad <= x and kvad <= y:
            break

```

```

        print("Uneta dimenzija ne odgovara, uneti ponovo!")
    return kvad

def inicijalizacija(x, y, kvad):
    M = [[0 for j in range(y)] for i in range(x)]
    N = [[0 for j in range(kvad)] for i in range(kvad)]
    return M, N

def popunjavanje_osnovne_matrice(x, y, M):
    for i in range(x):
        for j in range(y):
            M[i][j] = int(input("A[" + str(i) + "][" + str(j) + "] = "))

    return M

def popunjavanje_izvedene_matrice(x, y, kvad, M, N):
    for i in range(x):
        if i < kvad:
            for j in range(y):
                if j < kvad:
                    N[i][j] = M[i][j]
                else:
                    break
        else:
            break
    return N

def prikaz_rezultata(M, N):
    print("Osnovna matrica: ")
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

    print("Izvedena matrica: ")
    for red in N:
        for kolona in red:
            print(str(kolona), "\t", end = " ")
        print()

def main():
    m, n = unos_dimenzije_matrice()
    p = unos_dimenzije_kvadratne_matrice(m, n)
    A, B = inicijalizacija(m, n, p)
    A = popunjavanje_osnovne_matrice(m, n, A)
    B = popunjavanje_izvedene_matrice(m, n, p, A, B)
    prikaz_rezultata(A, B)

main()

```

Uneti broj redova matrice: 4

Uneti broj kolona matrice: 5

Uneti dimenziju kvadratne matrice: 3

A[0][0] = 0

A[0][1] = 0

A[0][2] = 0

A[0][3] = 0

A[0][4] = 0

A[1][0] = 1

A[1][1] = 1

A[1][2] = 1

A[1][3] = 1

A[1][4] = 1

A[2][0] = 2

A[2][1] = 2

A[2][2] = 2

A[2][3] = 2

A[2][4] = 2

A[3][0] = 3

A[3][1] = 3

A[3][2] = 3

A[3][3] = 3

A[3][4] = 3

0 0 0 0 0

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

0 0 0 0 0

1 1 1 1 1

2 2 2 2 2

Uneti broj redova matrice: 4

Uneti broj kolona matrice: 3

Uneti dimenziju kvadratne matrice: 3

A[0][0] = 0

A[0][1] = 0

A[0][2] = 0

A[1][0] = 1

A[1][1] = 1

A[1][2] = 1

A[2][0] = 2

A[2][1] = 2

A[2][2] = 2

A[3][0] = 3

A[3][1] = 3

A[3][2] = 3

Osnovna matrica:

0 0 0

1 1 1

2 2 2

3 3 3

Izvedena matrica:

0 0 0

1 1 1

2 2 2

6) Написати програм који коришћењем функција смештених у модуле попуњава матрицу димензије $m \times n$, приказује матрицу као и суме по колонама и редовима матрице.