

Задаци са матрицама**0112 Креирање матрице функцијама**

Написати програм који коришћењем функција креира матрицу реда 4 x 3, попуњава је са 2 а затим и приказује матрицу.

```
def kreiranje_matrice(x, y):
    M = [[2 for j in range(y)] for i in range(x)]
    return M
```

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

```
def main():
    A = kreiranje_matrice(4, 3)
    prikaz_rezultata(A)
```

```
main()
```

**0113 Сумирање елемената матрице**

Написати програм који коришћењем функција креира матрицу реда 3 x 2, попуњава први ред са 0, други са 1 а трећи са 2, затим и приказује матрицу и укупан збир свих елемената матрице.

```
def kreiranje_matrice():
    M = [[0, 0], [1, 1], [2, 2]]
    return M
```

```
def racunanje_sume_elementa_matrice(M):
    suma = 0
    for red in M:
        for kolona in red:
            suma += kolona
    return suma
```

```
def prikaz_rezultata(M, suma):
    for red in M:
        for kolona in red:
            print(str(kolona), "\t", end = "")
        print()
    print("Ukupan zbir elemenata je", suma)
```

```
def main():
    A = kreiranje_matrice()
    zbir = racunanje_sume_elementa_matrice(A)
    prikaz_rezultata(A, zbir)
```

```
main()
```

### 0114 Корисничко креирање матрице

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()
```

### Питања и задаци за самосталан рад

#### Задаци

038 Написати програм који коришћењем функција креира квадратну матрицу реда 3, попуњава је са потребном вредности а затим и приказује матрицу.

039 Написати програм који коришћењем функција креира матрицу реда  $2 \times 3$ , попуњава прву колону са 0, другу са 1 а трећу са 2, затим и приказује матрицу и укупан збир свих елемената матрице.

040 Написати програм који коришћењем функција креира матрицу  $[[3, 4], [7, 0], [2, 2]]$ , приказује матрицу и суме појединачних редова матрице.

### 0115 Рачунање суме матрице по редовима

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()
```

### 0116 Пребројавање елемената матрице

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()

```

#### Питања и задаци за самосталан рад

##### Задаци

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

042 Написати програм који помоћу функција попуњава матрицу димензије  $3 \times 3$  а затим креира нову матрицу истих димензија која прву колону прве матрице има као последњу колону.

#### **0117 Креирање матрице непознатих димензија**

Написати програм којим се уносе целобројне вредности у матрицу непознатих димензија

```

A = []
b = []
unos = input("Uneti n za nov red, vrednost za element matrice, q za kraj: ")
while unos != "q":
    while unos != "n":
        b.append(int(unos))
        unos = input("Uneti: ")
        if unos == "q" or unos == "n":
            break

A.append(b)

```

```
b = []
unos = input("Uneti: ")
```

```
print(A)
```

### 0118 Уређење матрице по дијагоналама

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
```

**0119 Сумирање матрице по дијагоналама**

Написати програм за израчунавање сума елемената матрице који се налазе изнад и посебне сума 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**

#### Питања и задаци за самосталан рад

##### Задаци

043 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.

044 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.

045 Матрицу  $[[\text{"p"}, \text{"a"}, \text{"j"}], [\text{"c"}, \text{"o"}, \text{"t"}], [\text{"i"}, \text{"o"}, \text{"o"}], [\text{"a"}, \text{"l"}, \text{"n"}], [\text{"n"}, \text{"e"}, \text{"j"}]]$  прочитати као спиралну матрицу.