

```

=====
def VectPrint(b, n):
#-----
# Печать элементов вектора b[1:n] на экран
#-----
    for i in range(1,n+1):
        print('%12.3e' % (b[i]),end='')
    print()
=====
def MatrPrint(a, n, m):
#-----
# Печать элементов матрицы a[1:n][1:m] на экран
#-----
    for i in range(1,n+1):
        for j in range (1,m+1): print('%12.3e' % (a[i][j]),end='')
        print()
=====
def MatrWrite(a, n, m, f):
#-----
# Печать элементов матрицы a[1:n][1:m] в файл
#-----
    for i in range(1,n+1):
        for j in range (1,m+1): print('%12.3e' % (a[i][j]),end='',file=f)
        print(file=f)
=====
def Resid(a, b, x, n):
#-----
# Расчет невязки R=max(abs(A*x-b)) решения СЛАУ A*x=b
#-----
    Rmax=0.0
    for i in range(1,n+1):
        R=-b[i]
        for j in range (1,n+1):
            R += a[i][j]*x[j] # невязка i-го уравнения
        Rmax=max(Rmax,abs(R))
    return Rmax
=====
def NRazn(x, y, n):
#-----
# Расчет нормы разности max(abs(x-y)) векторов x и y
#-----
    Rmax=0.0
    for i in range(1,n+1):
        R=x[i]-y[i]
        Rmax=max(Rmax,abs(R))
    return Rmax
=====
def MatrVec(a, b, n):
    """
    Расчет произведения матрицы a[1:n][1:n] на вектор b[1:n]
    """
    c = [0]*(n+1)
    for i in range(1,n+1):
        c[i] = 0.0

```

```

        for j in range (1,n+1):
            c[i] += a[i][j]*b[j]
    return c
#=====
def Gauss1(a, b, n):
    """
    Решение СЛАУ  $a \cdot x = b$  методом Гаусса с выбором ведущего элемента.
    a - матрица системы ((a11,...,a1n),...,(an1,...,ann)) (вход),
    преобразованная матрица (выход)
    b - вектор правой части (b1,...,bn) (вход), вектор решения x (выход)
    det - определитель матрицы a (выход).
    """
    det=1e0
    for k in range(1,n+1): # Прямой ход
        amax = 0e0 # определяем ведущую строку
        for i in range(k,n+1): # с максимальным элементом в столбце k
            if amax < abs(a[i][k]): amax = abs(a[i][k]); imax = i
        if amax == 0e0:
            print("Gauss1: singular matrix !"); return 0e0

        if imax != k: # меняем строки imax и k
            det = -det # чтобы поместить ведущий элемент на диагональ
            for j in range(k,n+1): (a[imax][j],a[k][j]) = (a[k][j],a[imax][j])
            (b[imax],b[k]) = (b[k],b[imax])

        det *= a[k][k] # домножаем определитель на ведущий элемент

        t = 1e0/a[k][k] # делим ведущую строку на ведущий элемент
        for j in range(k+1,n+1): a[k][j] *= t
        b[k] *= t

        for i in range(k+1,n+1): # преобразуем неведущие строки
            t = a[i][k]
            for j in range(k+1,n+1): a[i][j] -= a[k][j]*t
            b[i] -= b[k]*t

    for k in range(n-1,0,-1): # Обратный ход
        sum = b[k]
        for i in range(k+1,n+1): sum -= a[k][i]*b[i]
        b[k] = sum
    return b,det

```