

```

1 program questao1
2   implicit none
3   real, dimension(1000) :: x, y
4   integer :: n, i
5   real :: a, b, h, integra
6
7   open(file='dados.dat', unit=1, status='old')
8
9   read (1, *) n
10
11   do i = 1, n
12     read(1,*) x(i), y(i)
13   end do
14
15   print *, "O valor do deslocamento médio do automóvel é:"
16   print *, integra(n, x, y)
17
18
19 end program questao1
20
21 function integra(n, x, y) result(result)
22   implicit none
23   real, dimension(1000) :: x, y
24   real :: a, b, h, aux
25   real :: result
26   integer :: i, n
27
28   a = x(1)
29   b = x(n)
30   h = (b-a)/n
31
32   aux = 0.0
33   do i = 2, (n-1)
34     aux = aux+y(i)
35   end do
36
37   result = (h/2)*(y(1)+2*aux+y(n))
38 end function integra
39

```

```

1 program questao2
2   implicit none
3   real, dimension(3, 3) :: A
4   real, dimension(3) :: b, x
5   integer :: n, i
6
7   n = 3
8
9   data (A(1,i), i=1,3) /  3.0,  2.0,  -1.0 /
10  data (A(2,i), i=1,3) /  1.0,  3.0,  1.0 /
11  data (A(3,i), i=1,3) /  2.0,  2.0,  -2.0 /
12
13  data (b(i), i=1,3) /  0.0,  1.0,  2.0 /
14
15  call jordan(n, A, b, x)
16
17  print *, "Resultado do sistema de equações lineares"
18  do i = 1, 3
19    print *, x(i)
20  end do
21 end program questao2
22
23 subroutine jordan(n, A, b, x)
24   implicit none
25   real, dimension(3, 3) :: A
26   real, dimension(3) :: b, x
27   real :: m
28   integer :: i, j, k, n
29
30   do k = 1, n
31     do i = 1, n
32       if ( i .ne. k ) then
33         m = A(i, k)/A(k, k)
34         A(i, k) = 0.0
35         do j = k+1, n
36           A(i, j) = A(i, j) - m*A(k, j)
37         end do
38         b(i) = b(i) - m*b(k)
39       end if
40     end do
41   end do
42
43   do i = 1, n
44     x(i) = b(i)/A(i, i)
45   end do
46 end subroutine jordan

```

```

1 program questao3
2   implicit none
3   real :: V, L, R, i0, t0, i, t, tf, step
4   real, external :: didt
5
6   t0 = 0
7   i0 = 0
8
9   t = t0
10  i = i0
11
12  tf = 1
13  step = 1E-3
14
15
16  print *, "Insira o valor da tensão CC"
17  read(*,*) V
18  print *, "Insira o valor da indutância do indutor"
19  read(*,*) L
20  print *, "Insira o valor da resistência elétrica"
21  read(*,*) R
22
23  call euler(didt, V, L, R, i, t, tf, step)
24
25  print *, "O valor da corrente elétrica no tempo final:"
26  print *, i
27
28 end program questao3
29
30 subroutine euler(didt, V, L, R, i, t, tf, step)
31   implicit none
32   real :: didt, V, L, R, i, t, tf, step
33
34   do while(t<tf)
35     i = didt(V, L, R, i, t)*step + i
36     t = t + step
37   end do
38
39 end subroutine euler
40
41 function didt(V, L, R, i, t) result(result)
42   implicit none
43   real :: V, L, R, i, t
44   real :: result
45
46   result = (V-R*i)/L
47
48 end function didt

```