```
from fenics import*

mesh = UnitSquareMesh(8,8)
V = FunctionSpace(mesh,'P', 1)

u_D = Expression ('1 +x[0]*x[0]+2*x[1]*x[1]', degree = 2)

def boundary (x, on_boundary):
    return on_boundary

bc = DirichletBC(V,u_D,boundary)

u = TrialFunction (V)
v = TestFunction(V)
f = Constant(-6.0)
a = dot(grad(u), grad(v))*dx
L = f*v*dx

u = Function(V)

solve (a == L, u, bc)

vtkfile = File("poisson/solution.pvd")
vtkfile << u

error_L2 = errornorm(u_D,u,"L2")

vertex_values_u_D = u_D.compute_vertex_values(mesh)
vertex_values_u = u.compute_vertex_values(mesh)

import numpy as np
error_max = np.max(np.abs(vertex_values_u_D - vertex_values_u))

print("error_L2 =", error_L2)
print("error_max =", error_max)
```