```
%%%% Differential Equations Homework 4 - Romeo Perlstein %%%%

%%%% Useful notation:
%%%% heaviside function u(t), u_c(t-c)
```

Question E14

a)

y'' + 4y = (1 - u(t-2pi))sin(t), y(0) = 0, y'(0)=0

```
syms t s y(t) Y
og_eq_A = diff(y, 2) + 4*y == (1-heaviside(t-(2*pi)))*sin(t) % The OG equation
laplace_eq_A = laplace(equation_a, t, s) % The equation, not transformed using
 the Laplace transformation

better_laplace_eq_A = subs(eq1, [y(0), subs(diff(y(t), t), t, 0),
 laplace(y(t), t, s)], [0, 0, Y])
% uhhh not going to lie this is just what the textbook suggested. It
% apparently makes it easier to solve for y? my understanding is that,
% since eq1 had a bunch of subs values in it, this new format allows for
% us to simply do all that subbing in one line.

solve_laplace_eq_A = solve(better_laplace_eq_A, Y)
```

```
og_eq_A(t) =

4*y(t) + diff(y(t), t, t) == -sin(t)*(heaviside(t - 2*pi) - 1)


laplace_eq_A =

s^2*laplace(y(t), t, s) - s*y(0) - subs(diff(y(t), t), t, 0) + 4*laplace(y(t),
 t, s) == (exp(2*pi*s) - 1)/(exp(2*pi*s) + s^2*exp(2*pi*s))


better_laplace_eq_A =

Y*s^2 + 4*Y == (exp(2*pi*s) - 1)/(exp(2*pi*s) + s^2*exp(2*pi*s))


solve_laplace_eq_A =

(exp(2*pi*s) - 1)/((s^2 + 4)*(exp(2*pi*s) + s^2*exp(2*pi*s)))
```

Published with MATLAB® R2022b