Zadanie 3

Michał Łukomski 255552

```julia
using Plots
```

```julia
import PlotlyJS
```

```
PlotlyJSBackend()
```

```julia
plotlyjs()
```

```
interpolation (generic function with 2 methods)
```

```julia
interpolation(x, q_A, q_B, C=0) = x * q_A + (1 - x) * q_B + x * (1 - x) * C
```

```
unknown_mass_CsSiI3 = 0.082
```

```julia
unknown_mass_CsSiI3 = 0.5 * (0.095 + 0.069)
```

```
CsPbI₃ =
 (Eg = 1.73, Δ = 1.44, γ₁ = 9.1, γ₂ = 3.6, γ₃ = 0.7, mₕ = 0.095, Ep = 41.6, a = 6.238, α =
```

```julia
CsPbI₃ = @NamedTuple{Eg, Δ, γ₁, γ₂, γ₃, mₕ, Ep, a, α}((
    1.73, # Eg eV
    1.44, # Δ
    9.1, # γ₁
    3.6, # γ₂
    0.7, # γ₃
    0.095, # mₕ
    41.6, # Ep
    6.238, # a
    0.9, # α meV/K
))
```

```
CsSiI₃ =
  (Eg = 0.31, Δ = 0.5, γ₁ = 24.3, γ₂ = 11.5, γ₃ = 8.1, mₕ = 0.082, Ep = 18.9, a = 5.892, α =
```

```julia
CsSiI₃ = @NamedTuple{Eg, Δ, γ₁, γ₂, γ₃, mₕ, Ep, a, α}((
    0.31, # Eg
    0.50, # Δ
    24.3, # γ₁
    11.5, # γ₂
    8.1, # γ₃
    unknown_mass_CsSiI3, # mₕ
    18.9, # Ep
    5.892, # a
    0.1, # α meV/K
))
```

```
CsPbₓSi₁₋ₓI₃ (generic function with 2 methods)
```

```julia
CsPbₓSi₁₋ₓI₃(x, C=0) = @NamedTuple{Eg, Δ, γ₁, γ₂, γ₃, mₕ, Ep, a, α}(
    interpolation.(x, collect(CsPbI₃), collect(CsSiI₃), C)
)
```

```
x = 0.5
```

```julia
x = 0.5
```

```
material =
  (Eg = 1.02, Δ = 0.97, γ₁ = 16.7, γ₂ = 7.55, γ₃ = 4.4, mₕ = 0.0885, Ep = 30.25, a = 6.065,
```

```julia
material = CsPbₓSi₁₋ₓI₃(x)
```

```
Eg (generic function with 1 method)
```

```julia
Eg(T, mat) = mat.Eg + mat.α * 1e-3 * T
```

```
VB₀ = 0.0
```

```julia
VB₀ = 0.0 # eV
```

```
VB (generic function with 1 method)
```

```julia
VB(T, mat) = VB₀
```

```
CS (generic function with 1 method)
```

```julia
CS(T, mat) = VB₀ + Eg(T, mat)
```

```
CH (generic function with 1 method)
```

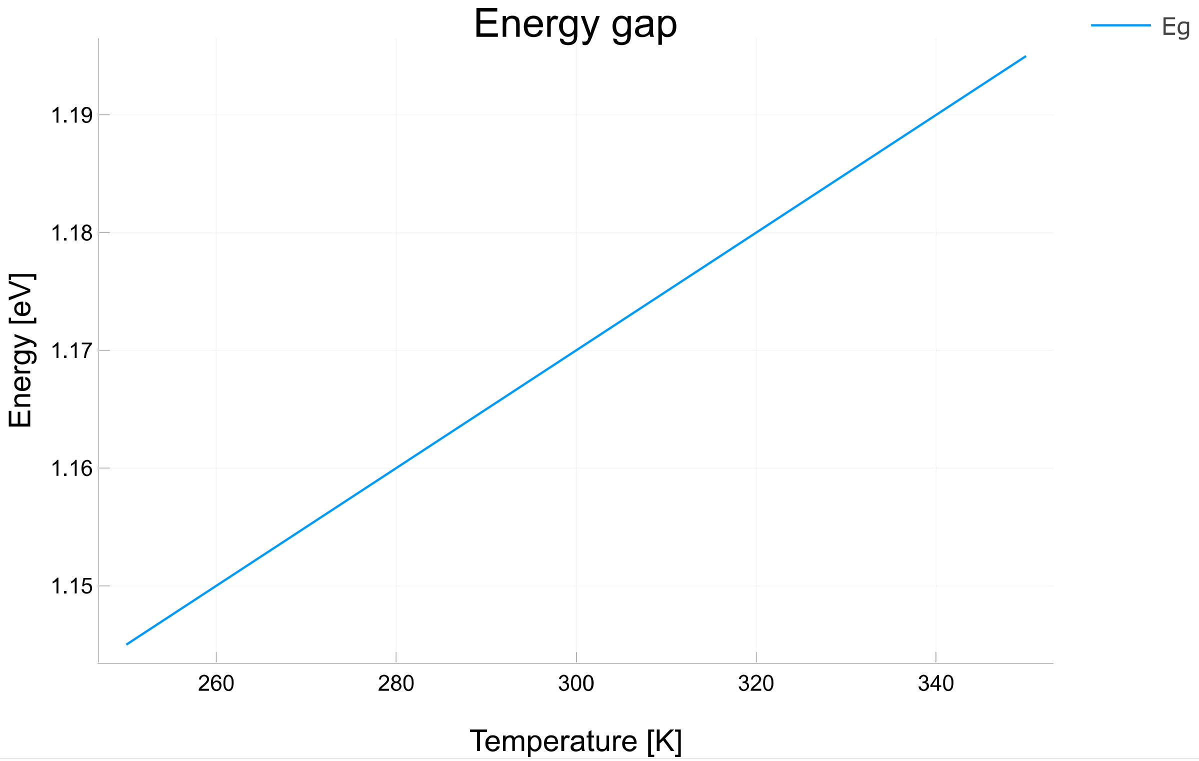
```julia
CH(T, mat) = VB₀ + Eg(T, mat) + mat.Δ
```

```
CL (generic function with 1 method)
```

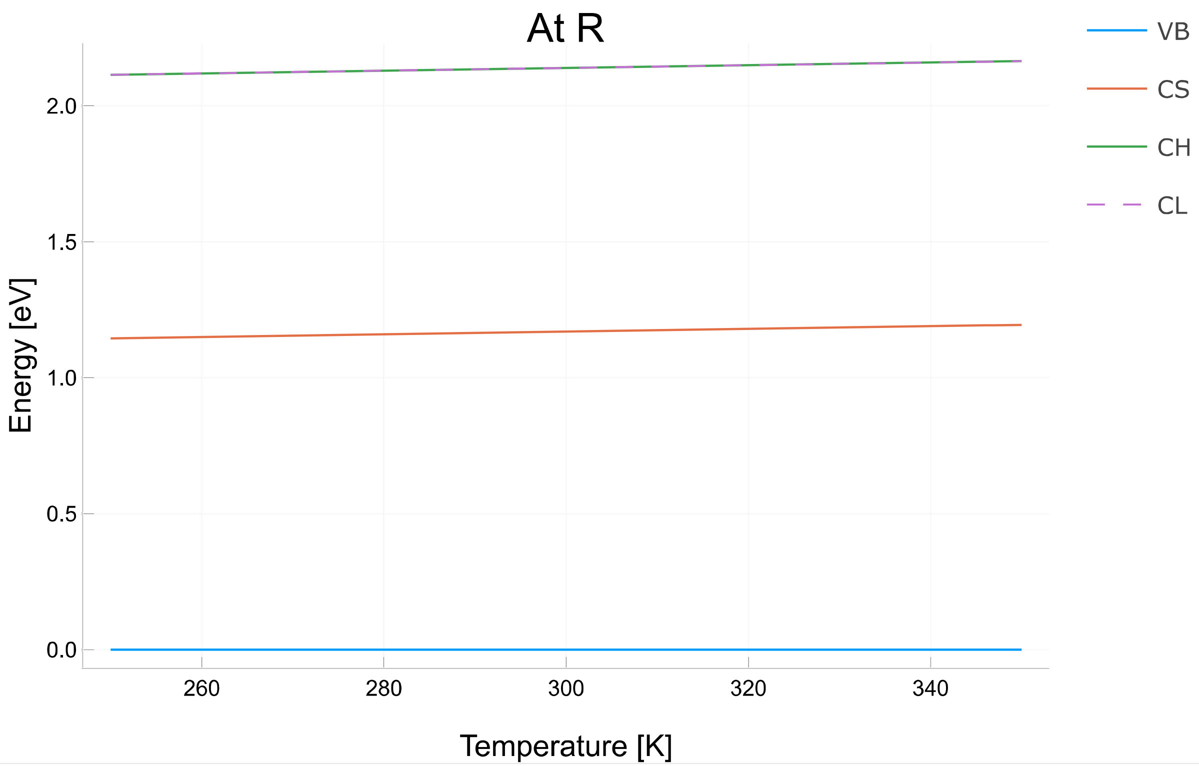
```julia
CL(T, mat) = CH(T, mat)
```

```
Ts = 250:350
```

```julia
Ts = 250:350 # K
```



```
plot(Ts, T->Eg(T,material), label="Eg",
    xlabel="Temperature [K]",
    ylabel="Energy [eV]",
    title="Energy gap"
)
```



```
begin
    plot(Ts, T -> VB(T, material), label="VB",
        xlabel="Temperature [K]",
        ylabel="Energy [eV]",
        title="At R"
    )
    # plot(Ts, VB.(Ts, Ref(material)))
    plot!(Ts, T -> CS(T, material), label="CS")
    plot!(Ts, T -> CH(T, material), label="CH")
    plot!(Ts, T -> CL(T, material), label="CL", linestyle=:dash)
end
```