PARTITION FUNCTION

Before moving to the practical let's see the basic terms and definition for proper understanding:

Fermi-Dirac Statistics:

Fermi-Dirac statistics is used to describe the behaviour of fermions, which are the particles that obey the Pauli exclusion principle, meaning no two fermions can occupy the same quantum state simultaneously. The probability distribution function for the occupation of different energy states by fermions at a given temperature T is given by:

$$n_i = \frac{1}{e^{\frac{\varepsilon_i - \mu}{k_B T}} + 1}$$

Where:
- n_i is the average number of particles in the i^{th} energy state.
- ε_i is the energy of the i^{th} state.
- μ is the chemical potential.
- k_B is the Boltzmann constant.
- T is the temperature.

The partition function Z for a system of fermions is given by the sum over all possible energy states ε_i by the Fermi-Dirac distribution:

$$Z = \sum_i e^{-\frac{\varepsilon_i - \mu}{k_B T}}$$

Maxwell Boltzmann Statistics:

Maxwell-Boltzmann statistics describe the behaviour of classical particles, such as non-interacting gas particles. In this case, there is no restriction on the number of particles occupying a particular energy state. The probability distribution function for the occupation of different energy states by classical particles at a given temperature T is given by the Boltzmann distribution:

$$n_i = \frac{1}{e^{\frac{\varepsilon_i}{k_B T}}}$$

Where:
- n_i is the average number of particles in the i^{th} energy state.
- ε_i is the energy of the i^{th} state.

- k_B is the Boltzmann constant.
- T is the temperature

For Maxwell-Boltzmann statistics, the partition function is the same as the denominator in the Boltzmann distribution:

$$Z = \sum_i e^{-\frac{\varepsilon_i}{k_B T}}$$

Bose-Einstein Statistics:

Bose-Einstein statistics is used to describe the behaviour of bosons, which are the particles that follow Bose-Einstein distribution. Bosons do not obey Pauli's Exclusion principle, meaning any number of bosons can occupy the same quantum state. The probability distribution function for the occupation of different energy states by bosons at a given temperature (T) is given by:

$$n_i = \frac{1}{e^{\frac{\varepsilon_i - \mu}{k_B T}} - 1}$$

Where:
- n_i is the average number of particles in the i^{th} energy state.
- ε_i is the energy of the i^{th} state.
- μ is the chemical potential.
- k_B is the Boltzmann constant.
- T is the temperature

The partition function Z for a system of bosons is given by the sum over all possible energy states ε_i by the Bose-Einstein distribution:

$$Z = \sum_i e^{-\frac{\varepsilon_i - \mu}{k_B T}}$$

Average Energy:

The average energy $\langle E \rangle$ of the system can be calculated using the partition function:

$$\langle E \rangle = -\frac{1}{Z}\frac{\partial Z}{\partial \beta}$$

Where $\beta = \frac{1}{k_B T}$.

Specific Heat:

The specific heat C_V at constant volume is given by:

$$C_V = \frac{\partial \langle E \rangle}{\partial T}$$

Helmholtz Free Energy:

The Helmholtz free energy F is given by:

$$F = -k_B T \ln(Z)$$

Entropy:

The entropy S is given by:

$$S = \frac{\langle E \rangle - F}{T}$$

EXPERIMENT 1

AIM: To study the partition function for two bosons present in three energy levels of energy 0J, 100J, 200J using Bose Einstein Statistics and the plot the various graphs showing variation of:

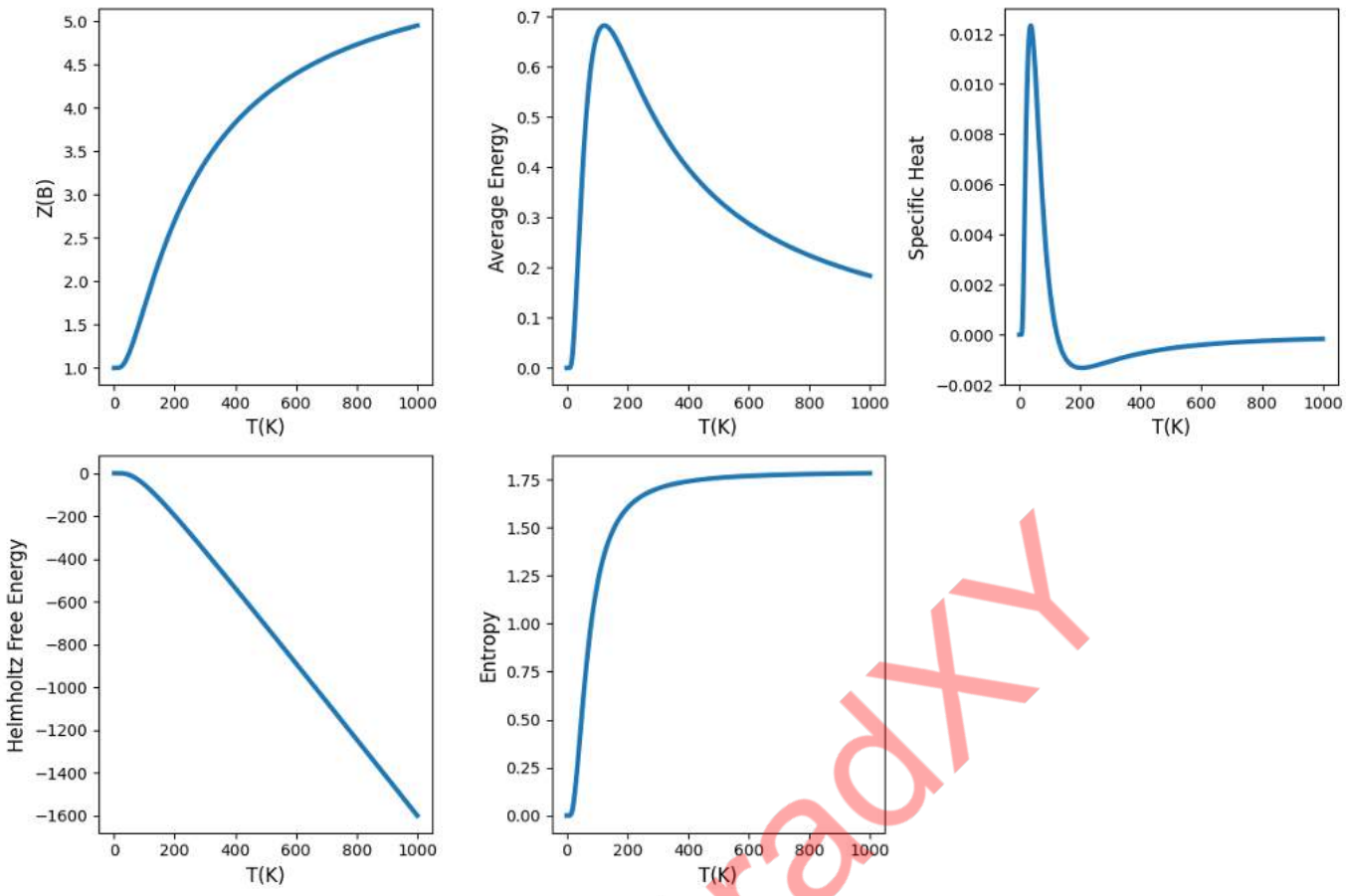
- Partition function with temperature
- Average Energy with temperature
- Specific Heat with temperature
- Helmoltz free energy with temperature
- Entropy with temperature

CODE:

```
import numpy as np
import matplotlib.pyplot as plt

k = 1  # Boltzmann constant
E1 = 0;E2 = 100;E3 = 200
N = 2  # Number of bosons
dT = 0.1  # Temperature interval

T = np.arange(0.1, 1000 + dT, dT)
T1 = T[:-1]  # For first derivative with respect to temperature
T2 = T1[:-1]  # For second derivative with respect to temperature
# Loop for partition function
Z = np.exp(-2*E1/T) + np.exp(-2*E2/T) + np.exp(-2*E3/T) + np.exp((-E1-E2)/T) +
np.exp((-E2-E3)/T) + np.exp((-E1-E3)/T)
lnZ = np.log(Z)

U = k * T1 * np.diff(lnZ) / dT # Average Energy with Temperature
Cv = np.diff(U) / dT # Specific Heat with Temperature
```

```
F = -k * T * lnZ # Helmholtz free energy with Temperature
S = -np.diff(F) / dT # Entropy with Temperature

# Plotting the graphs
plt.figure(figsize=(12, 8))

plt.subplot(2, 3, 1) # Partition function with Temperature
plt.plot(T, Z, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Z(B)", fontsize=12)

plt.subplot(2, 3, 2) # Average Energy with Temperature
plt.plot(T1, U, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Average Energy", fontsize=12)

plt.subplot(2, 3, 3) # Specific Heat with Temperature
plt.plot(T2, Cv, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Specific Heat", fontsize=12)

plt.subplot(2, 3, 4) # Helmholtz free energy with Temperature
plt.plot(T, F, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Helmholtz Free Energy", fontsize=12)

plt.subplot(2, 3, 5) # Entropy with Temperature
plt.plot(T1, S, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Entropy", fontsize=12)

plt.tight_layout()
plt.show()
```


OUTPUT:



EXPERIMENT 2

AIM: To study the partition function for two particles, present in three energy levels of energy 0J, 100J, 200J using Maxwell Boltzmann Statistics and the plot the various graphs showing variation of:

- Partition function with temperature
- Average Energy with temperature
- Specific Heat with temperature
- Helmoltz free energy with temperature
- Entropy with temperature

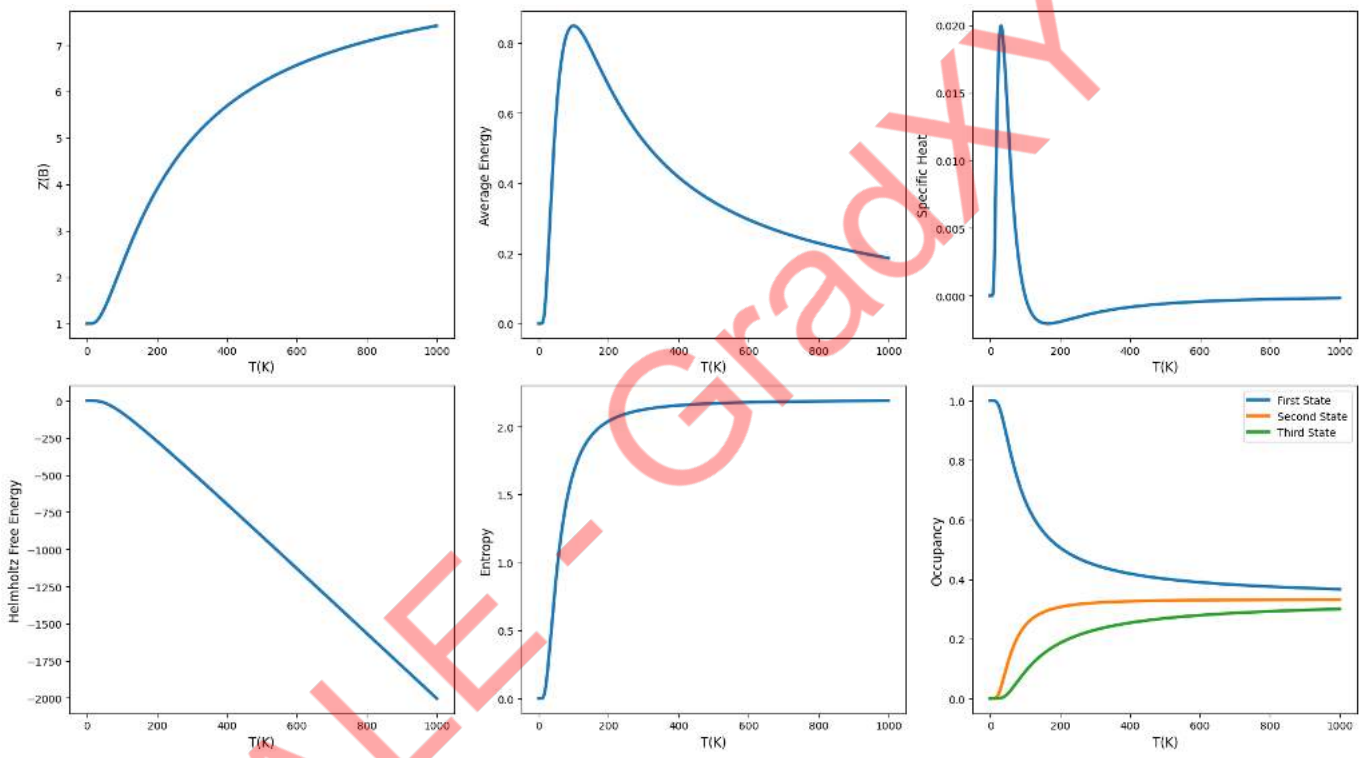
CODE:

```
import numpy as np
import matplotlib.pyplot as plt

k = 1  # Boltzmann constant
E1 = 0;E2 = 100;E3 = 200
N = 2  # Number of particles
dT = 0.1  # Temperature interval
T = np.arange(0.1, 1000 + dT, dT)
T1 = T[:-1]  # For first derivative with respect to temperature
```

```
T2 = T1[:-1]  # For second derivative with respect to temperature

# Initialize arrays
z1 = np.exp(-E1 / T)
z2 = np.exp(-E2 / T)
z3 = np.exp(-E3 / T)
z = z1 + z2 + z3
Z = z**N

# Occupancy probabilities
p1 = z1 / z
p2 = z2 / z
p3 = z3 / z

lnZ = np.log(Z) #Natural logarithm of the partition function

U = k * T1 * np.diff(lnZ) / dT # Average Energy with Temperature
Cv = np.diff(U) / dT # Specific Heat with Temperature
F = -k * T * lnZ # Helmholtz free energy with Temperature
S = -np.diff(F) / dT # Entropy with Temperature

# Plotting the graphs
plt.figure(figsize=(18, 10))

plt.subplot(2, 3, 1) # Partition function with Temperature
plt.plot(T, Z, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Z(B)", fontsize=12)

plt.subplot(2, 3, 2) # Average Energy with Temperature
plt.plot(T1, U, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Average Energy", fontsize=12)

plt.subplot(2, 3, 3) # Specific Heat with Temperature
plt.plot(T2, Cv, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Specific Heat", fontsize=12)

plt.subplot(2, 3, 4) # Helmholtz free energy with Temperature
plt.plot(T, F, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Helmholtz Free Energy", fontsize=12)

plt.subplot(2, 3, 5) # Entropy with Temperature
plt.plot(T1, S, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Entropy", fontsize=12)
```

```
plt.subplot(2, 3, 6) # Occupancy of States with Temperature
plt.plot(T, p1, label="First State", linewidth=3)
plt.plot(T, p2, label="Second State", linewidth=3)
plt.plot(T, p3, label="Third State", linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Occupancy", fontsize=12)
plt.legend()

plt.tight_layout()
plt.show()
```

OUTPUT:



EXPERIMENT 3

AIM: To study the partition function for two fermions present in three energy levels of energy 0J, 100J, 200J using Fermi Dirac Statistics and the plot the various graphs showing variation of:

- Partition function with temperature
- Average Energy with temperature
- Specific Heat with temperature
- Helmoltz free energy with temperature
- Entropy with temperature

CODE:

```
import numpy as np
import matplotlib.pyplot as plt

k = 1  # Boltzmann constant
E1 = 0;E2 = 100;E3 = 200
N = 2  # Number of fermions
dT = 0.1  # Temperature interval

T = np.arange(0.1, 1000 + dT, dT)
T1 = T[:-1]  # For first derivative with respect to temperature
T2 = T1[:-1]  # For second derivative with respect to temperature

# Calculate partition function Z(T)
Z = np.exp((-E1 - E2) / T) + np.exp((-E2 - E3) / T) + np.exp((-E1 - E3) / T)
lnZ = np.log(Z) #Natural logarithm of the partition function

U = k * T1**2 * np.diff(lnZ) / dT # Average Energy U(T)
Cv = np.diff(U) / dT # Specific Heat Cv(T)
F = -k * T * lnZ # Helmholtz Free Energy F(T)
S = -np.diff(F) / dT # Entropy S(T)

# Plotting the graphs
plt.figure(figsize=(18, 10))

plt.subplot(2, 3, 1) # Partition function with Temperature
plt.plot(T, Z, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Z(B)", fontsize=12)

plt.subplot(2, 3, 2) # Average Energy with Temperature
plt.plot(T1, U, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Average Energy", fontsize=12)
plt.subplot(2, 3, 3) # Specific Heat with Temperature
plt.plot(T2, Cv, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Specific Heat", fontsize=12)

plt.subplot(2, 3, 4) # Helmholtz Free Energy with Temperature
plt.plot(T, F, linewidth=3)
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Helmholtz Free Energy", fontsize=12)

plt.subplot(2, 3, 5) # Entropy with Temperature
plt.plot(T1, S, linewidth=3)
```

```
plt.xlabel("T(K)", fontsize=12)
plt.ylabel("Entropy", fontsize=12)

plt.tight_layout()
plt.show()
```

OUTPUT: