Name : Sanjay AR **Roll No :** CB.SC.U4CSE23052
Date : 5th January 2025 **Set :** 2

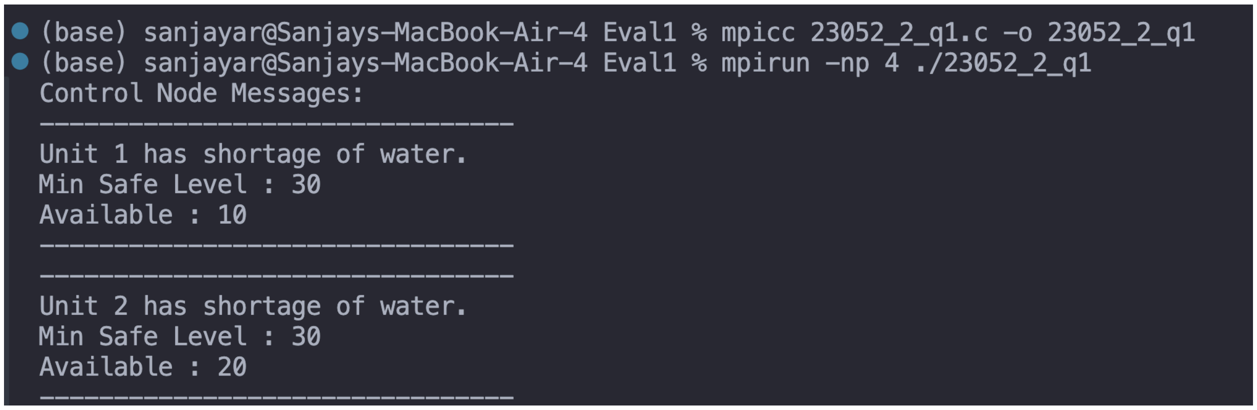
Q1) Each reservoir control unit maintains water level data. All units must know the minimum available level to prevent drought risk.

Code

```
#include <stdio.h>
#include <mpi.h>

/*
Q1: Each reservoir control unit maintains water level data.
All units must know the minimum available level to prevent drought risk.
*/

int main(int argc, char **argv)
{

   int rank, size, waterLevel, minLevel;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);

   // Set Min Level to 30
   if (rank == 0)
       minLevel = 30;

   // Broadcast min level
   MPI_Bcast(&minLevel, 1, MPI_INT, 0, MPI_COMM_WORLD);

   if (rank != 0)
   {

       // water level data
       waterLevel = rank * 10;

       if (waterLevel < minLevel)
       {
```

```c
            MPI_Send(&waterLevel, 1, MPI_INT, 0, rank, MPI_COMM_WORLD);
        }

        else
        {
            int msg = -1;
            MPI_Send(&msg, 1, MPI_INT, 0, rank, MPI_COMM_WORLD);
        }
    }

    else if (rank == 0)
    {
        MPI_Status status;
        printf("Control Node Messages:\n");
        for (int i = 0; i < size - 1; i++)
        {
            MPI_Recv(&waterLevel, 1, MPI_INT, MPI_ANY_SOURCE, MPI_ANY_TAG,
MPI_COMM_WORLD, &status);

            if (waterLevel == -1)
                continue;

            printf("--------------------------------\n");
            printf("Unit %d has shortage of water.\n", status.MPI_SOURCE);
            printf("Min Safe Level : %d\n", minLevel);
            printf("Available : %d\n", waterLevel);
            printf("--------------------------------\n");
        }
    }

    MPI_Finalize();
    return 0;
}
```

Output


```
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpicc 23052_2_q1.c -o 23052_2_q1
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpirun -np 4 ./23052_2_q1
Control Node Messages:
--------------------------------
Unit 1 has shortage of water.
Min Safe Level : 30
Available : 10
--------------------------------
--------------------------------
Unit 2 has shortage of water.
Min Safe Level : 30
Available : 20
--------------------------------
```

```
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpicc 23052_2_q1.c -o 23052_2_q1
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpirun -np 4 ./23052_2_q1
Control Node Messages:
--------------------------------
Unit 1 has shortage of water.
Min Safe Level : 30
Available : 10
--------------------------------
--------------------------------
Unit 2 has shortage of water.
Min Safe Level : 30
Available : 20
--------------------------------
```

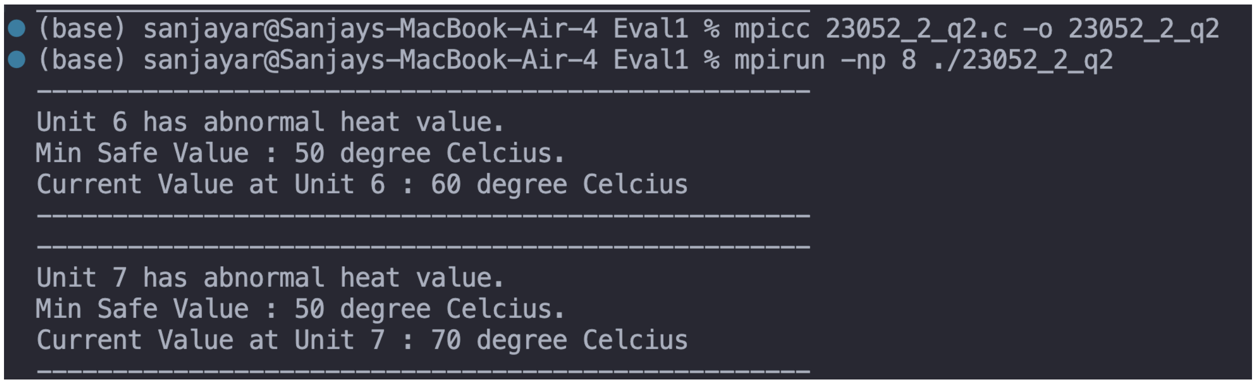
Q2) An industrial plant has several sensor units monitoring temperature. If a sensor detects abnormal heat, its value must be sent to the control unit.

<u>Code</u>

```c
#include <stdio.h>
#include <mpi.h>

/*
Q2:  An industrial plant has several sensor units monitoring temperature.
If a sensor detects abnormal heat, its value must be sent to the control unit.
*/

int main(int argc, char **argv)
{

   int rank, size, currHeat, safeHeat;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);

   // Bcast safe heat values to all sensor units from control unit
   // Setting safe heat to 50
   if (rank == 0)
       safeHeat = 50;

   MPI_Bcast(&safeHeat, 1, MPI_INT, 0, MPI_COMM_WORLD);

   if (rank != 0)
   {

       // Set currHeat for each sensor unit
       currHeat = rank * 10;
       // MPI_Send(&currHeat, 1, MPI_INT, 0, rank, MPI_COMM_WORLD);

       // Detect abnormal heat and report to control unit
       if (currHeat > safeHeat)
       {
           MPI_Send(&currHeat, 1, MPI_INT, 0, rank, MPI_COMM_WORLD);
       }

       // else send placeholder message
```

```
        else
        {
            int msg = -1;
            MPI_Send(&msg, 1, MPI_INT, 0, rank, MPI_COMM_WORLD);
        }
    }

    else if (rank == 0)
    {
        MPI_Status status;
        for (int i = 0; i < size - 1; i++)
        {
            MPI_Recv(&currHeat, 1, MPI_INT, MPI_ANY_SOURCE, MPI_ANY_TAG,
MPI_COMM_WORLD, &status);

            if (currHeat == -1)
                continue;

            printf("---------------------------------------------------\n");
            printf("Unit %d has abnormal heat value.\n", status.MPI_SOURCE);
            printf("Min Safe Value : %d degree Celcius.\n", safeHeat);
            printf("Current Value at Unit %d : %d degree Celcius\n", status.MPI_SOURCE,
currHeat);
            printf("---------------------------------------------------\n");
        }
    }

    MPI_Finalize();
    return 0;
}
```

Output


```
● (base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpicc 23052_2_q2.c -o 23052_2_q2
● (base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpirun -np 8 ./23052_2_q2
  ---------------------------------------------------
  Unit 6 has abnormal heat value.
  Min Safe Value : 50 degree Celcius.
  Current Value at Unit 6 : 60 degree Celcius
  ---------------------------------------------------
  ---------------------------------------------------
  Unit 7 has abnormal heat value.
  Min Safe Value : 50 degree Celcius.
  Current Value at Unit 7 : 70 degree Celcius
  ---------------------------------------------------
```

```
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpicc 23052_2_q2.c -o 23052_2_q2
(base) sanjayar@Sanjays-MacBook-Air-4 Eval1 % mpirun -np 8 ./23052_2_q2
---------------------------------------------------
Unit 6 has abnormal heat value.
Min Safe Value : 50 degree Celcius.
Current Value at Unit 6 : 60 degree Celcius
---------------------------------------------------
---------------------------------------------------
Unit 7 has abnormal heat value.
Min Safe Value : 50 degree Celcius.
Current Value at Unit 7 : 70 degree Celcius
---------------------------------------------------
```