Q1

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

int main(int argc, char** argv) {
   int rank, size;
   int x=2; // You can change this constant to any integer val;
   // Initialize the MPI environment
   MPI_Init(&argc, &argv);

   // Get the rank of the process
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);

   // Get the number of processes
   MPI_Comm_size(MPI_COMM_WORLD, &size);

   // Calculate pow(x, rank)
   double result = pow(x, rank);

   // Print the result
   printf("Process %d: %f\n", rank, result);

   // Finalize the MPI environment
   MPI_Finalize();

   return 0;
}
```

Q2

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

```
#include <stdio.h>

int main(int argc, char** argv) {
   // Initialize the MPI environment
   MPI_Init(NULL, NULL);

   // Get the rank of the process
   int world_rank;
   MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

   // Check if the rank is even or odd and print the corresponding message
   if (world_rank % 2 == 0) {
      printf("Hello from process %d\n", world_rank);
   } else {
      printf("World from process %d\n", world_rank);
   }

   // Finalize the MPI environment.
   MPI_Finalize();
}
```

Q3

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

int main(int argc, char *argv[]) {
   int rank, size;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);
```

```
double a = 5.0;
double b = 2.0;
double result;

switch(rank) {
   case 0:
      result = a + b;
      printf("Process %d - Addition: %f\n", rank, result);
      break;
   case 1:
      result = a - b;
      printf("Process %d - Subtraction: %f\n", rank, result);
      break;
   case 2:
      result = a * b;
      printf("Process %d - Multiplication: %f\n", rank, result);
      break;
   case 3:
      if(b != 0) {
         result = a / b;
         printf("Process %d - Division: %f\n", rank, result);
      } else {
         printf("Process %d - Division by zero error\n", rank);
      }
      break;
   default:
      printf("Process %d - No operation assigned\n", rank);
      break;
}
```

```
    MPI_Finalize();
    return 0;
}
```

Q4:

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

void toggle_case(char *c) {
    if (islower(*c)) {
        *c = toupper(*c);
    } else if (isupper(*c)) {
        *c = tolower(*c);
    }
}

int main(int argc, char *argv[]) {
    int rank, size;
    char str[] = "HeLLO";
    int str_len = strlen(str);

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

    if (rank < str_len) {
        toggle_case(&str[rank]);
    }
```

```
    char result[str_len];
    MPI_Gather(&str[rank], 1, MPI_CHAR, result, 1, MPI_CHAR, 0, MPI_COMM_WORLD);

    if (rank == 0) {
        printf("Toggled string: %s\n", result);
    }

    MPI_Finalize();
    return 0;
}
```