

C q1.c

• C q2.c

C q3.c

C q4.c

Week4 > C q1.c > main(int, char *[])

```
1  #include <mpi.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  long long factorial(int n) {
6      long long fact = 1;
7      for (int i = 1; i <= n; i++)
8          fact *= i;
9      return fact;
10 }
11
12 int main(int argc, char *argv[]) {
13     int rank, size;
14     int err;
15     char err_string[MPI_MAX_ERROR_STRING];
16     int err_len;
17     long long local_fact;
18     long long scan_sum;
19
20     err = MPI_Init(&argc, &argv);
21     if (err != MPI_SUCCESS) {
22         MPI_Error_string(err, err_string, &err_len);
23         printf("%s\n", err_string);
24         MPI_Abort(MPI_COMM_WORLD, err);
25     }
26
27     MPI_Comm_set_errhandler(MPI_COMM_WORLD, MPI_ERRORS_RETURN);
28
29     err = MPI_Comm_rank(MPI_COMM_WORLD, &rank);
30     if (err != MPI_SUCCESS) {
31         MPI_Error_string(err, err_string, &err_len); if (rank == 0) {
32             printf("Q1, Adarsh Ranjan 230962278\n");
33         }
34
35         printf("%s\n", err_string);
36         MPI_Abort(MPI_COMM_WORLD, err);
37     }
38
39     err = MPI_Comm_size(MPI_COMM_WORLD, &size);
40     if (err != MPI_SUCCESS) {
41         MPI_Error_string(err, err_string, &err_len);
42         printf("%s\n", err_string);
43         MPI_Abort(MPI_COMM_WORLD, err);
44     }
45
46     local_fact = factorial(rank + 1);
47
48     err = MPI_Scan(&local_fact, &scan_sum, 1, MPI_LONG_LONG, MPI_SUM, MPI_COMM_WORLD);
49     if (err != MPI_SUCCESS) {
50         MPI_Error_string(err, err_string, &err_len);
51         printf("%s\n", err_string);
52         MPI_Abort(MPI_COMM_WORLD, err);
53     }
54
55     if (rank == size - 1) {
56         printf("Final Output: %lld\n", scan_sum);
57     }
58     if (rank == 0) {
59         printf("Q1, Adarsh Ranjan 230962278\n");
60     }
61
62     MPI_Finalize();
63     return 0;
64 }
65
```

```

• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpicc q1.c -o q1
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpirun -np 5 ./q1
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Final Output: 153
Q1, Adarsh Ranjan 230962278

```

```

C q1.c  C q2.c  X  C q3.c  C q4.c
Week4 > C q2.c > main(int, char* [])
1  #include <mpi.h>
2  #include <stdio.h>
3
4  int main(int argc, char *argv[]) {
5      int rank, size, err;
6      char err_string[MPI_MAX_ERROR_STRING];
7      int err_len;
8      int matrix[9], local_count=0, total_count=0, search;
9      int chunk[3];
10
11      err=MPI_Init(&argc,&argv);
12      if(err!=MPI_SUCCESS){MPI_Error_string(err,err_string,&err_len);printf("%s\n",err_string);MPI_Abort(MPI_COMM_WORLD,e
13      MPI_Comm_set_errhandler(MPI_COMM_WORLD,MPI_ERRORS_RETURN);
14
15      MPI_Comm_rank(MPI_COMM_WORLD,&rank);
16      MPI_Comm_size(MPI_COMM_WORLD,&size);
17
18
19      if(rank==0){
20          printf("Enter the elements \n");
21
22          for(int i=0;i<9;i++) scanf("%d",&matrix[i]);
23          printf("Enter element to search: \n");
24          scanf("%d",&search);
25      }
26
27      MPI_Bcast(&search,1,MPI_INT,0,MPI_COMM_WORLD);
28      MPI_Scatter(matrix,3,MPI_INT,chunk,3,MPI_INT,0,MPI_COMM_WORLD);
29
30      for(int i=0;i<3;i++)
31          if(chunk[i]==search) local_count++;
32
33      MPI_Reduce(&local_count,&total_count,1,MPI_INT,MPI_SUM,0,MPI_COMM_WORLD);
34
35      if(rank==0) printf("Total Count:%d\n",total_count);
36      if(rank == 0) {
37          printf("Q2, Adarsh Ranjan 230962278\n");
38      }
39
40      MPI_Finalize();
41      return 0;
42  }
43

```

```
Q2, Adarsh Ranjan 230962278
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpicc q2.c -o q2
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpirun -np 3 ./q2
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Enter the elements
1
2
3
4
3
5
6
4
3
Enter element to search:
3
Total Count:3
Q2, Adarsh Ranjan 230962278
```

```

C q1.c  C q2.c  C q3.c  X  C q4.c
Week4 > C q3.c > main(int, char *[])
1  #include <mpi.h>
2  #include <stdio.h>
3
4  int main(int argc, char *argv[]) {
5      int rank, size, err;
6      char err_string[MPI_MAX_ERROR_STRING];
7      int err_len;
8      int matrix[16], row[4], result[16];
9
10     err=MPI_Init(&argc,&argv);
11     if(err!=MPI_SUCCESS){
12         MPI_Error_string(err,err_string,&err_len);
13         printf("%s\n",err_string);
14         MPI_Abort(MPI_COMM_WORLD,err);
15     }
16
17     MPI_Comm_set_errhandler(MPI_COMM_WORLD,MPI_ERRORS_RETURN);
18
19     MPI_Comm_rank(MPI_COMM_WORLD,&rank);
20     MPI_Comm_size(MPI_COMM_WORLD,&size);
21
22     if(rank==0){
23         printf("Enter elements of 4x4 matrix:\n");
24         for(int i=0;i<16;i++)
25             scanf("%d",&matrix[i]);
26     }
27
28     MPI_Scatter(matrix,4,MPI_INT,row,4,MPI_INT,0,MPI_COMM_WORLD);
29
30     for(int i=0;i<4;i++)
31         row[i]+=rank;
32
33     MPI_Gather(row,4,MPI_INT,result,4,MPI_INT,0,MPI_COMM_WORLD);
34
35     if(rank==0){
36         printf("Output matrix:\n");
37         for(int i=0;i<16;i++){
38             printf("%d ",result[i]);
39             if((i+1)%4==0) printf("\n");
40         }
41     }
42     if(rank == 0) {
43         printf("Q3, Adarsh Ranjan 230962278\n");
44     }
45
46     MPI_Finalize();
47     return 0;
48 }
49

```

```
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpicc q3.c -o q3
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpirun -np 4 ./q3
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Enter elements of 4x4 matrix:
1
2
3
4
1
2
3
1
1
1
1
1
2
1
2
1
Output matrix:
1 2 3 4
2 3 4 2
3 3 3 3
5 4 5 4
Q3, Adarsh Ranjan 230962278
```

```

C q1.c  C q2.c  C q3.c  C q4.c  X
Week4 > C q4.c > main(int, char* [])
1  #include <mpi.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  int main(int argc, char *argv[]) {
6      int rank, size, err;
7      char err_string[MPI_MAX_ERROR_STRING];
8      int err_len;
9      char word[100], ch;
10     char result[1000];
11     int counts[100], displs[100];
12     char local[100];
13
14     err=MPI_Init(&argc,&argv);
15     if(err!=MPI_SUCCESS){
16         MPI_Error_string(err, err_string, &err_len);
17         printf("%s\n", err_string);
18         MPI_Abort(MPI_COMM_WORLD, err);
19     }
20     MPI_Comm_set_errhandler(MPI_COMM_WORLD, MPI_ERRORS_RETURN);
21
22     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
23     MPI_Comm_size(MPI_COMM_WORLD, &size);
24
25     if(rank==0){
26         printf("Enter the word:\n");
27         scanf("%s", word);
28     }
29
30     MPI_Scatter(word, 1, MPI_CHAR, &ch, 1, MPI_CHAR, 0, MPI_COMM_WORLD);
31
32     for(int i=0; i<=rank; i++) local[i]=ch;
33     local[rank+1]='\0';
34
35     for(int i=0; i<size; i++){
36         counts[i]=i+1;
37         displs[i]=i*(i+1)/2;
38     }
39
40     MPI_Gatherv(local, rank+1, MPI_CHAR, result, counts, displs, MPI_CHAR, 0, MPI_COMM_WORLD);
41
42     if(rank==0) printf("Output word:\n%s\n", result);
43     if(rank == 0) {
44         printf("Q4, Adarsh Ranjan 230962278\n");
45     }
46
47     MPI_Finalize();
48     return 0;
49 }
50
51

```

```

• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpicc q4.c -o q4
• (base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week4$ mpirun -np 4 ./q4
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Enter the word:
PCAP
Output word:
PCCAAAPPPP
Q4, Adarsh Ranjan 230962278

```