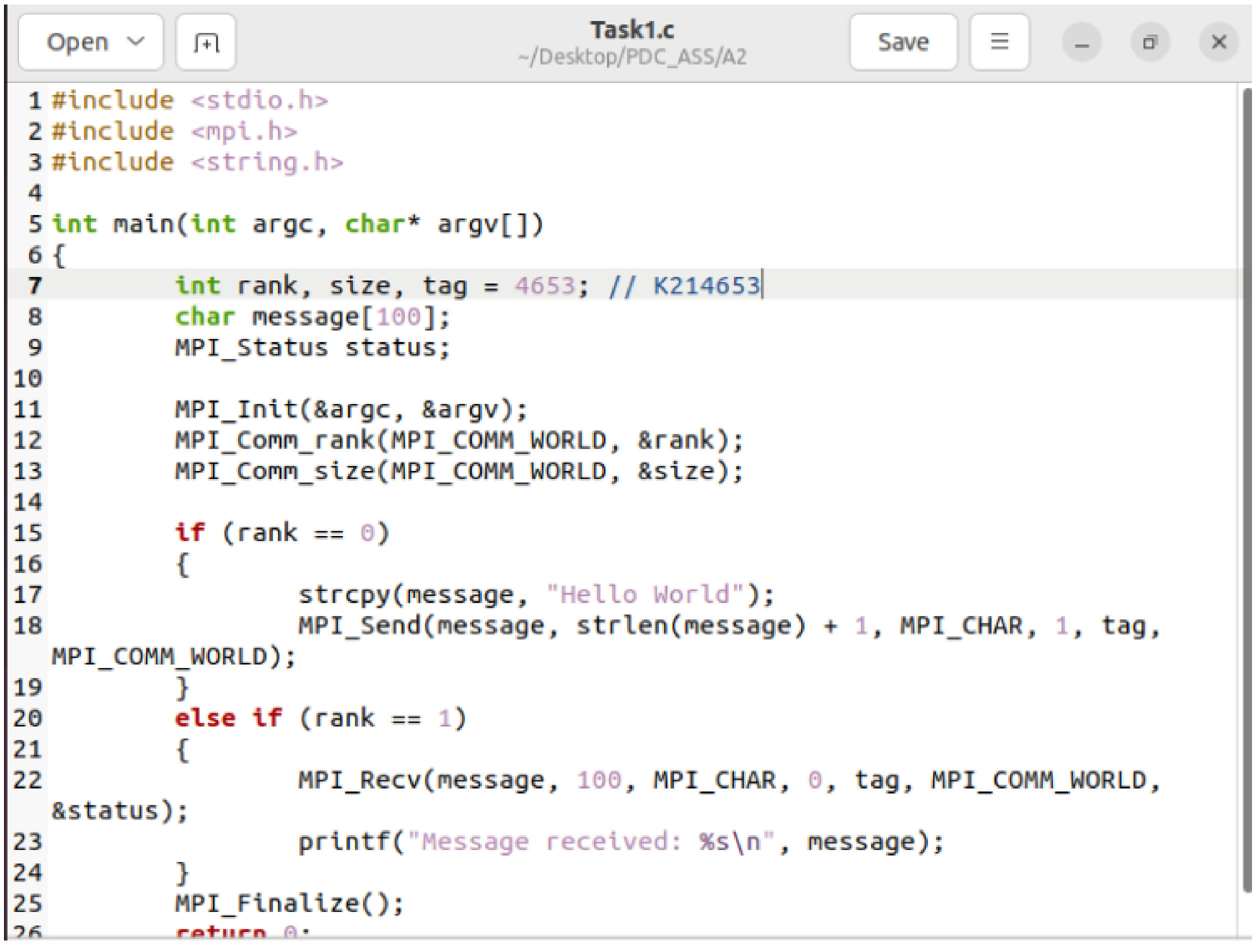
PDC A2

Ali Zain 21K4653

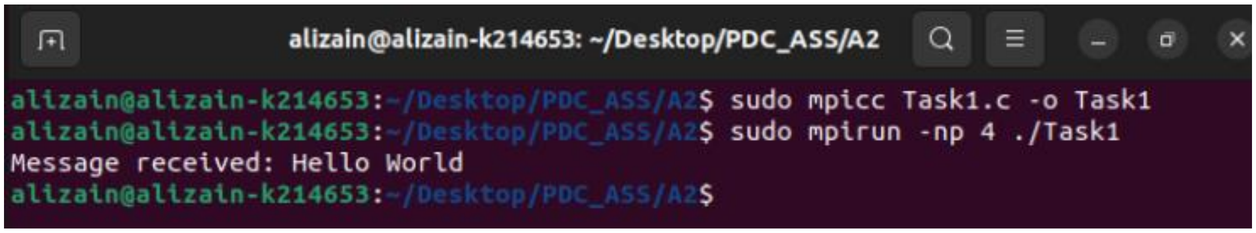
Task #1:

Code:



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

int main(int argc, char* argv[])
{
	int rank, size, tag = 4653; // K214653
	char message[100];
	MPI_Status status;

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

	if (rank == 0)
	{
		strcpy(message, "Hello World");
		MPI_Send(message, strlen(message) + 1, MPI_CHAR, 1, tag,
MPI_COMM_WORLD);
	}
	else if (rank == 1)
	{
		MPI_Recv(message, 100, MPI_CHAR, 0, tag, MPI_COMM_WORLD,
&status);
		printf("Message received: %s\n", message);
	}
	MPI_Finalize();
	return 0;
```

Output:



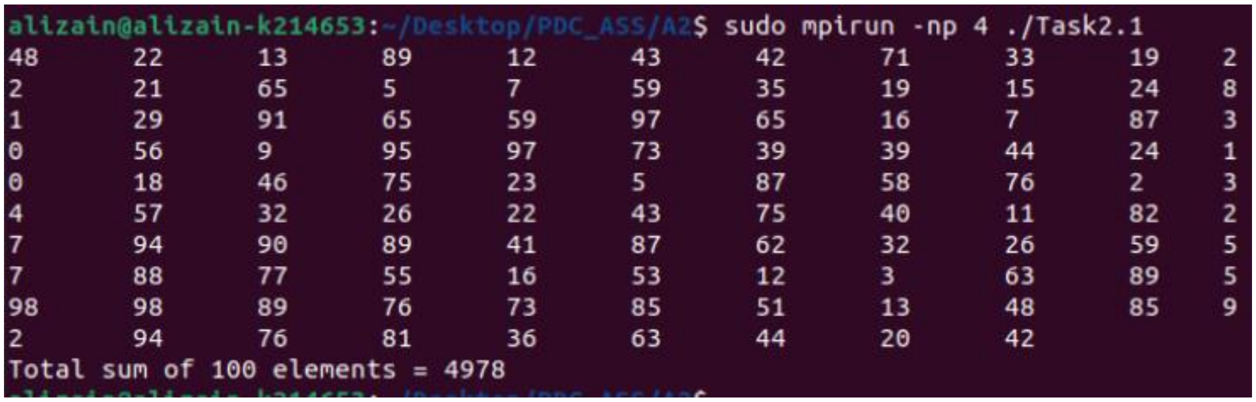
```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpicc Task1.c -o Task1
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 4 ./Task1
Message received: Hello World
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$
```

Task #2:

Code:

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

#define N 100

int main(int argc, char* argv[]) {
    int rank, size;
    time_t t;
    int num[N] = { 0 };
    int* arr = NULL, * arr2 = NULL;
    int sum = 0, total_sum = 0;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    srand((unsigned)time(&t));
    if (rank == 0) {
        for (int i = 0; i < N; i++) {
            num[i] = rand() % 100;
            printf("%d\t", num[i]);

        }
    }

    arr = (int*)malloc(N / size * sizeof(int));
    MPI_Scatter(num, N / size, MPI_INT, arr, N / size, MPI_INT, 0, MPI_COMM_WORLD);

    for (int i = 0; i < N / size; i++) {
        sum += arr[i];
    }


    if (rank == 0) {
        arr2 = (int*)malloc(size * sizeof(int));
    }
    MPI_Gather(&sum, 1, MPI_INT, arr2, 1, MPI_INT, 0, MPI_COMM_WORLD);


    if (rank == 0) {
        for (int i = 0; i < size; i++) {
            total_sum += arr2[i];
        }
        printf("\nTotal sum of 100 elements = %d\n", total_sum);
    }
    MPI_Finalize();

    return 0;
}
```

Output:



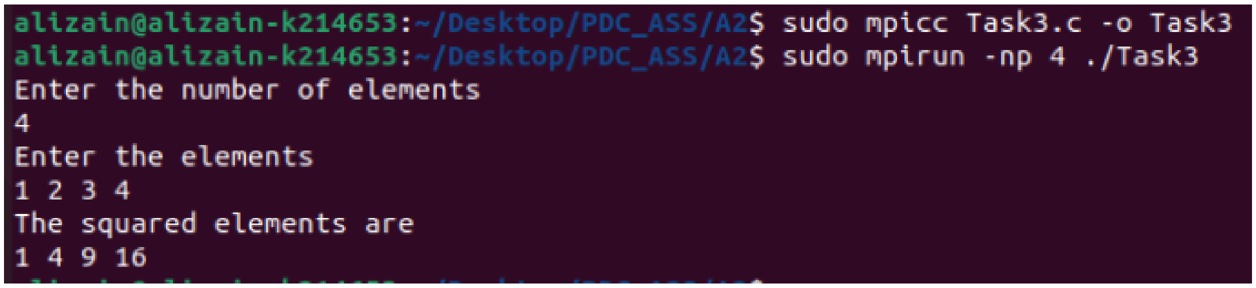
```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 4 ./Task2.1
48      22      13      89      12      43      42      71      33      19      2
2       21      65      5       7       59      35      19      15      24      8
1       29      91      65      59      97      65      16      7       87      3
0       56      9       95      97      73      39      39      44      24      1
0       18      46      75      23      5       87      58      76      2       3
4       57      32      26      22      43      75      40      11      82      2
7       94      90      89      41      87      62      32      26      59      5
7       88      77      55      16      53      12      3       63      89      5
98      98      89      76      73      85      51      13      48      85      9
2       94      76      81      36      63      44      20      42
Total sum of 100 elements = 4978
```

Task #3:

Code:

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

int main(int argc, char* argv[])
{
        int rank, size, i, n, * a, * b, * c, * d;
        MPI_Init(&argc, &argv);
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
        MPI_Comm_size(MPI_COMM_WORLD, &size);
        if (rank == 0)
        {
                printf("Enter the number of elements\n");
                scanf("%d", &n);
        }
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
        a = (int*)malloc(n * sizeof(int));
        b = (int*)malloc(n * sizeof(int));
        c = (int*)malloc(n * sizeof(int));
        d = (int*)malloc(n * sizeof(int));
        if (rank == 0)
        {
                printf("Enter the elements\n");
                for (i = 0; i < n; i++)
                        scanf("%d", &a[i]);
        }
        MPI_Scatter(a, 1, MPI_INT, b, 1, MPI_INT, 0, MPI_COMM_WORLD);
        for (i = 0; i < n; i++)
                c[i] = b[i] * b[i];
        MPI_Gather(c, 1, MPI_INT, d, 1, MPI_INT, 0, MPI_COMM_WORLD);
        if (rank == 0)
        {
                printf("The squared elements are\n");
                for (i = 0; i < n; i++)
                        printf("%d ", d[i]);
                printf("\n");
        }
        MPI_Finalize();
        return 0;
}
```

Output:



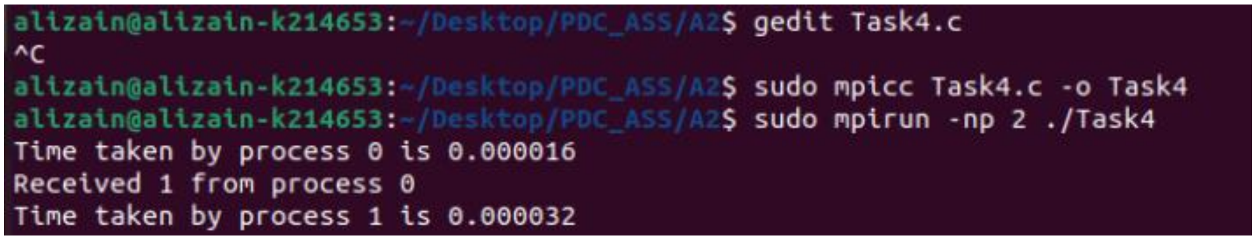
```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpicc Task3.c -o Task3
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 4 ./Task3
Enter the number of elements
4
Enter the elements
1 2 3 4
The squared elements are
1 4 9 16
```

Task #4:

Code:

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

int main(int argc, char* argv[])
{
	int rank, size, tag = 4653;
	double start, end;
	MPI_Status status;
	MPI_Init(&argc, &argv);
	MPI_Comm_rank(MPI_COMM_WORLD, &rank);
	start = MPI_Wtime();
	if (rank == 0)
	{
		int a = 10;
		MPI_Send(&a, 1, MPI_INT, 1, tag, MPI_COMM_WORLD);
	}
	else if (rank == 1)
	{
		int b;
		MPI_Recv(&b, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &status);
		printf("Received %d from process 0\n", b);
	}
	end = MPI_Wtime();
	printf("Time taken by process %d is %f\n", rank, end - start);
	MPI_Finalize();
	return 0;
}
```

Output:



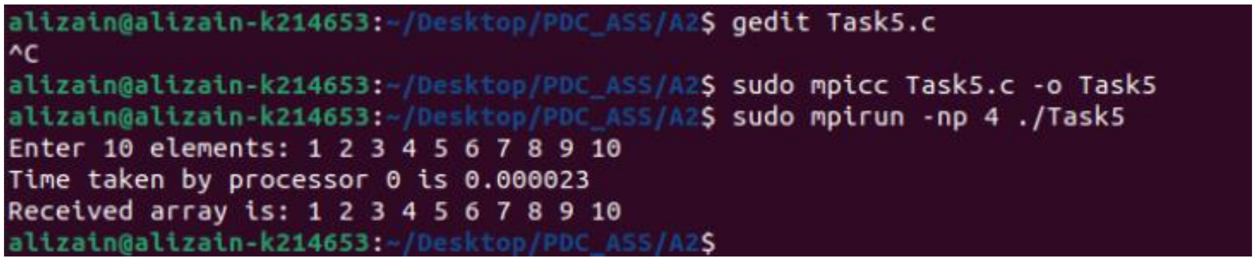
```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ gedit Task4.c
^C
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpicc Task4.c -o Task4
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 2 ./Task4
Time taken by process 0 is 0.000016
Received 1 from process 0
Time taken by process 1 is 0.000032
```

Task #5:

Code:

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

int main(int argc, char* argv[])
{
	int rank, size, i, n = 10;
	int* arr = (int*)malloc(n * sizeof(int));
	double start, end;
	MPI_Init(&argc, &argv);
	MPI_Comm_rank(MPI_COMM_WORLD, &rank);
	MPI_Comm_size(MPI_COMM_WORLD, &size); //size = 2
	if (rank == 0)
	{
		printf("Enter %d elements: ", n);
		for (i = 0; i < n; i++)
			scanf("%d", &arr[i]);
		start = MPI_Wtime();
		MPI_Send(arr, n, MPI_INT, 1, 0, MPI_COMM_WORLD);
		end = MPI_Wtime();
		printf("Time taken by processor %d is %f\n", rank, end - start);
	}
	else if (rank == 1)
	{
		MPI_Recv(arr, n, MPI_INT, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
		printf("Received array is: ");
		for (i = 0; i < n; i++)
			printf("%d ", arr[i]);
		printf("\n");
	}
	MPI_Finalize();
	return 0;
}
```

Output:



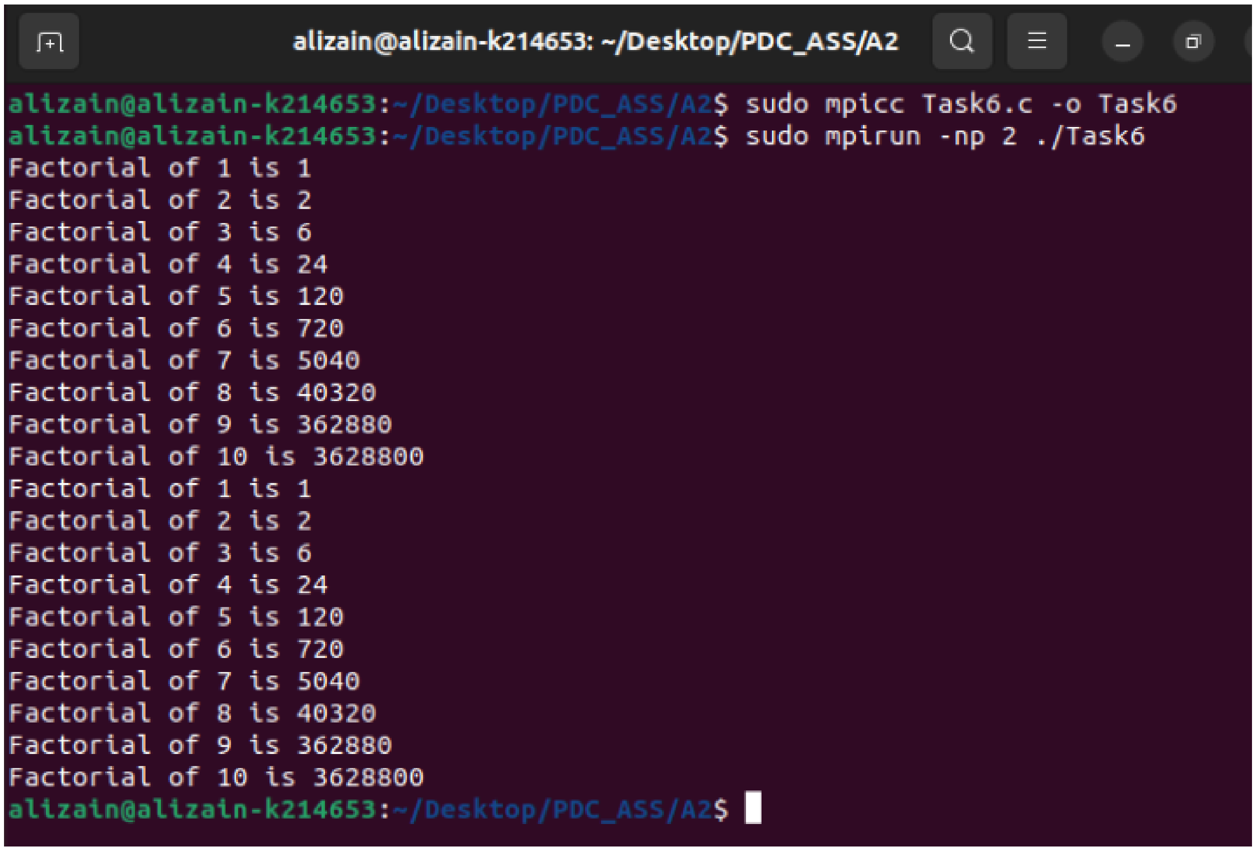
```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ gedit Task5.c
^C
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpicc Task5.c -o Task5
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 4 ./Task5
Enter 10 elements: 1 2 3 4 5 6 7 8 9 10
Time taken by processor 0 is 0.000023
Received array is: 1 2 3 4 5 6 7 8 9 10
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$
```

Task #6:

Code:

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

int main(int argc, char* argv[])
{
	int rank, size, fact = 1, i, j, n = 10, fact1 = 1;
	MPI_Init(&argc, &argv);
	MPI_Comm_rank(MPI_COMM_WORLD, &rank);
	MPI_Comm_size(MPI_COMM_WORLD, &size);

	if (rank == 0)
	{
		for (i = 1; i <= n; i++)
		{
			fact = fact * i;
			printf("Factorial of %d is %d\n", i, fact);
		}
	}
	MPI_Finalize();
	return 0;
}
```

Output: because of 2 processor 2 outputs each from different processor



```
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpicc Task6.c -o Task6
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$ sudo mpirun -np 2 ./Task6
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
Factorial of 8 is 40320
Factorial of 9 is 362880
Factorial of 10 is 3628800
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
Factorial of 8 is 40320
Factorial of 9 is 362880
Factorial of 10 is 3628800
alizain@alizain-k214653:~/Desktop/PDC_ASS/A2$
```