

```
scanf("%d", &num_rows_1);
printf("Enter the number of columns of first matrix\n");
scanf("%d", &num_cols_1);
printf("Enter the values row by row one on each line\n" );
for ( i = 0; i < num_rows_1; i++)
for( j=0; j<num_cols_1; j++)
{
scanf("%d", &matrix_1[i][j]);
}
size[0][0] = num_rows_1;
size[0][1] = num_cols_1;
printf("Enter the number of rows of second matrix\n");
scanf("%d", &num_rows_2);
printf("Enter the number of columns of second matrix\n");
scanf("%d", &num_cols_2);
if( num_cols_1 != num_rows_2)
{
printf("MATRICES CANNOT BE MULTIPLIED\n");
exit(1);
}
printf("Enter the values row by row one on each line\n");
for (i = 0; i < num_rows_2; i++)
for(j=0; j<num_cols_2; j++)
{
scanf("%d", &matrix_2[i][j]);
}
size[1][0] = num_rows_2;
size[1][1] = num_cols_2;
if((sock_fd = socket(AF_INET, SOCK_DGRAM, 0)) < 0)
{
printf("Cannot create socket\n");
exit(1);
}
bzero((char*)&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_port = htons(atoi(argv[2]));
inet_pton(AF_INET, argv[1], &servaddr.sin_addr);

// SENDING MATRIX WITH SIZES OF MATRICES 1 AND 2
n = sendto(sock_fd, size, sizeof(size), 0, (struct sockaddr*)&servaddr, sizeof(servaddr));
```