```cpp
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <string.h>
#include "mpi.h" // message passing interface
using namespace std;

//
// Program 3
// In Your EYE - 10 points
//  Rachel Burke
//

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

    int my_rank;       // my CPU number for this process
    int p;             // number of CPUs that we have
    int source;        // rank of the sender
    int dest;          // rank of destination
    int tag = 0;       // message number
    char message[100]; // message itself
    MPI_Status status; // return status for receive

    // Start MPI
    MPI_Init(&argc, &argv);

    // Find out my rank!
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

    // Find out the number of processes!
    MPI_Comm_size(MPI_COMM_WORLD, &p);

    // THE REAL PROGRAM IS HERE

    // Pseudo-random number generator seeded at 1251
    srand(1251);

    // Array to contain many letters and array to maintain letter counts
    int n = 100000;
    char *letters = new char[n];
    int letter_counts[26] = {0};

    //Create a textfile and read its data into the letters array
    if (my_rank == 0)
    {
        char *text = new char[n];
        for (int x = 0; x < n; x++)
            text[x] = (char)(rand() % 26 + 97);

        ofstream outfile("letters.txt");
        if (outfile.is_open())
        {
            outfile.write((char *)text, n);
            outfile.close();
        }
        else
            cout << "Error: No File." << endl;

        ifstream infile("letters.txt");
        infile >> letters;

        delete[] text;
    }

    // Divide the problem
    int local_n = n / p;
    char *local_array = new char[local_n];
```

```
70      int local_counts[26] = {0};
71
72      MPI_Scatter(&letters[0], local_n, MPI_CHAR, local_array, local_n, MPI_CHAR, 0,
        MPI_COMM_WORLD);
73
74      // Do the local work
75      for (int x = 0; x < local_n; x++)
76          local_counts[(int)local_array[x] - 97]++;
77
78      for (int x = 0; x < 26; x++)
79          MPI_Reduce(&local_counts[x], &letter_counts[x], 1, MPI_INT, MPI_SUM, 0,
            MPI_COMM_WORLD);
80
81      delete[] local_array;
82
83      // Print result
84      if (my_rank == 0)
85      {
86          //Counting the remaining letters that exist
87          for (int x = n - (n % p); x < n; x++)
88              letter_counts[(int)letters[x] - 97]++;
89
90          //Printing Counts and Total
91          int total_count = 0;
92          for (int x = 0; x < 26; x++)
93          {
94              cout << (char)(x + 97) << ": " << letter_counts[x] << endl;
95              total_count += letter_counts[x];
96          }
97          cout << "Total Letters: " << total_count << endl;
98      }
99
100     delete[] letters;
101
102     // Shut down MPI
103     MPI_Finalize();
104
105     return 0;
106 }
```