```
Script started on 2023-10-04 23:06:16+00:00 [TERM="xterm-256color" TTY="/dev/pts/4" COLUM
NS="57" LINES="37"]
\[\033[01;34m\]\w\[\033[00m\]$ pwd
/home/runner/Lab-10-A-Review-Problem-Version-2-kcp3s
\[\033[01;34m\]\w\[\033[00m\]$ ls -la
total 2588
drwxr-xr-x 1 runner runner     292 Oct  4 23:06 .
drwxrwxrwx 1 runner runner     130 Oct  4 23:04 ..
-rwxr-xr-x 1 runner runner   18008 Oct  4 23:05 a.out
-rw-r--r-- 1 runner runner      17 Aug 18 20:59 .breakpoints
drwxr-xr-x 1 runner runner      12 Jan 24  2022 .cache
drwxr-x--- 1 runner runner     592 Oct  3 17:53 .ccls-cache
drwxr-xr-x 1 runner runner      68 Oct  4 20:50 .lesson
-rwxr-xr-x 1 runner runner 1291624 Oct  4 00:44 main
-rw-r--r-- 1 runner runner    2668 Oct  4 20:52 main.cpp
-rwxr-xr-x 1 runner runner 1258728 Aug 18 20:58 main-debug
-rw-r--r-- 1 runner runner   29432 Apr 21  2022 main.o
-rw-r--r-- 1 runner runner     432 Aug 18 21:02 Makefile
-rw-r--r-- 1 runner runner       0 Oct  4 23:06 Patel_lab_10.log
-rw-r--r-- 1 runner runner    1426 Dec 21  2022 .replit
-rw-r--r-- 1 runner runner     143 Oct  3 13:58 replit.nix
-rw-r--r-- 1 runner runner      16 Oct  3 12:39 T0.dat
-rw-r--r-- 1 runner runner      28 Oct  3 12:39 T1.dat
-rw------- 1 runner runner     183 Oct  3 13:09 T2.dat
-rw------- 1 runner runner       0 Oct  3 12:40 T3.dat
-rw------- 1 runner runner       7 Oct  3 12:40 T4.dat
-rw------- 1 runner runner       7 Oct  3 12:40 T5.dat
\[\033[01;34m\]\w\[\033[00m\]$ cat -n main.cpp
     1  #include <array>
     2  #include <fstream>
     3  #include <iomanip>
     4  #include <iostream>
     5  #include <string>
     6
     7  std::array<int, 31> ProcessFile(std::ifstream &, int &);
     8  double CalculateFinalGrade(std::array<int, 31> grades, int, int);
     9  char CalculateLetter(double);
    10  int CalculateTotal(std::array<int, 31> grades, int);
    11
    12  int main() {
    13    std::string filename;
    14    int num_grades = 0;
    15    int count = 0;
    16
    17    // Getting user input
    18    std::cout << "Enter the input file: ";
    19    std::cin >> filename;
    20
    21    // Opening the file
    22    std::ifstream file;
    23    file.open(filename);
    24
    25    // If file does not exist
    26    if (!file) {
    27      std::cout << '\n' << filename << " does not exist.\n";
    28      return 0;
    29    }
    30    std::cout << "\n";
    31    // Assigning variables to the functions
    32    std::array<int, 31> grades = ProcessFile(file, num_grades);
    33    int max_points = num_grades * 100;
    34    double final_grade = CalculateFinalGrade(grades, max_points, num_grades);
    35    char letter_grade = CalculateLetter(final_grade);
    36    int total_grade = CalculateTotal(grades, num_grades);
    37    // Outputting content that we got from the functions
    38    std::cout << "Number of grades: " << std::setw(11) << num_grades << std::endl;
    39    std::cout << "Total Points Earned: " << std::setw(8) << total_grade
    40              << std::endl;
    41    std::cout << "Max Possible Points: " << std::setw(8) << max_points << std::endl
;
```

```
    42    std::cout << "\nFinal Grade: " << std::setw(7) << letter_grade << std::setw(8)
<< std::fixed
    43              << std::setprecision(1) << final_grade << "%"<< std::endl;
    44    // Closing the file
    45    file.close();
    46  }
    47
    48  // Defining function ProcessFile
    49  std::array<int, 31> ProcessFile(std::ifstream &file, int &num_grades) {
    50    std::array<int, 31> grades;
    51    int count = 0;
    52    while (file >> grades.at(count) && count < 30) {
    53      count++;
    54    }
    55    num_grades = count;
    56    return grades;
    57  }
    58
    59  // Defining function CalculateFinalGrade
    60  double CalculateFinalGrade(std::array<int, 31> grades, int max_points, int count)
 {
    61    static double final_grade;
    62    double total_points = CalculateTotal(grades, count);
    63    if (total_points == 0 || max_points == 0){
    64      return 0.0;
    65    }
    66     final_grade = (total_points / max_points) * 100;
    67    return final_grade;
    68  }
    69
    70  // Defining function CalculateLetter
    71  char CalculateLetter(double final_grade) {
    72    if (final_grade >= 90.0) {
    73      return 'A';
    74    } else if (final_grade >= 80.0) {
    75      return 'B';
    76    } else if (final_grade >= 70.0) {
    77      return 'C';
    78    } else if (final_grade >= 60.0) {
    79      return 'D';
    80    } else {
    81      return 'F';
    82    }
    83  }
    84
    85  // Defining function CalculateTotal
    86  int CalculateTotal(std::array<int, 31> grades, int count) {
    87    int total_points = 0;
    88    for (int i = 0; i < count; i++) {
    89      total_points += grades[i];
    90    }
    91    return total_points;
    92  }
\[\033[01;34m\]\w\[\033[00m\]$ g++ main.cpp -o review2
\[\033[01;34m\]\w\[\033[00m\]$ ./review2
Enter the input file: T0.dat

Number of grades:           5
Total Points Earned:      448
Max Possible Points:      500

Final Grade:       B    89.6%
\[\033[01;34m\]\w\[\033[00m\]$ ./review2
Enter the input file: T1.dat

Number of grades:          10
Total Points Earned:      318
Max Possible Points:     1000

Final Grade:       F    31.8%
```

```
\[\033[01;34m\]\w\[\033[00m\]$ ./review2
Enter the input file: T2.dat

Number of grades:          30
Total Points Earned:     2191
Max Possible Points:     3000

Final Grade:       C    73.0%
\[\033[01;34m\]\w\[\033[00m\]$ ./review2
Enter the input file: T3.dat

Number of grades:           0
Total Points Earned:        0
Max Possible Points:        0

Final Grade:       F     0.0%
\[\033[01;34m\]\w\[\033[00m\]$ ./review2
Enter the input file: random.dat

random.dat does not exist.
\[\033[01;34m\]\w\[\033[00m\]$ exit

Script done on 2023-10-04 23:08:20+00:00 [COMMAND_EXIT_CODE="0"]
```