Lab Report 7: Strings, Plotting and Recursion

Jashan Singh

Department of Engineering, Toronto Metropolitan University

CPS 188: Computer Programming Fundamentals

Dr. Denis Hamelin

21 March 2024

Section: 07, Student No.- 501226507

Solution 1: Algorithm:

1. Firstly in the code we have imported the important libraries that are required to run the code
2. Then important functions have been defined for code
3. After this in the main function the string has been asked for the user and after performing the calculations based on the functions defined by the algorithm, the string is cleaned and then reversed
4. Then it is checked whether the string is palindrome or not which means whether the strings that are reversed are equal to the original string or not.
5. Following this the result is printed to the user

Code:

```
//Here we have imported the important libraries that are required to run the
program
#include <stdio.h>
#include <ctype.h>
#include <string.h>
//Here we are defining the void functions to run the program here
void clean(char before[], char after[]);
void reverse(char before[], char after[]);
int is_palindrome(char str[]);
// This is the main function of the program
int main() {
    char original[1000], removed_spaces[1000], reversed[1000];
    printf("String that you want to check: ");
    fgets(original, sizeof(original), stdin);
    printf("Original string: %s", original);
    without_space(original, removed_spaces);
    printf("Cleaned string: %s\n", removed_spaces);
    reverse(removed_spaces, reversed);
    printf("Reversed string: %s\n", reversed);
    if (is_palindrome(removed_spaces))
        printf("Your provided string is a palindrome.\n");
    else
        printf("Your provided string is not a palindrome\n");
    return 0;
}
void without_space(char before[], char after[]) {
    int j = 0;
    for (int i = 0; before[i] != '\0'; i++) {
        if (isalnum(before[i])) {
            after[j] = tolower(before[i]);
            j++;
        }
    }
    after[j] = '\0';
}
//This is a void function for reversing the palindrome
```

```
void reverse(char before[], char after[]) {
    if (strlen(before) == 1) {
        after[0] = before[0];
        after[1] = '\0';
    } else {
        reverse(before + 1, after);
        int len = strlen(after);
        after[len] = before[0];
        after[len + 1] = '\0';
    }
}

//Here the conditions are defined for checking the palindrome
int is_palindrome(char str[]) {
    int len = strlen(str);
    for (int i = 0; i < len / 2; i++) {
        if (str[i] != str[len - i - 1]) {
            return 0;
        }
    }
    return 1;
}
```

Output:

```
/tmp/GdLMGZnCc4.o
String that you want to check: Drab as a fool, aloof as a bard.
Original string: Drab as a fool, aloof as a bard.
Cleaned string: drabasafoolaloofasabard
Reversed string: drabasafoolaloofasabard
Your provided string is a palindrome.
/tmp/s9Q7YwBVYV.o
String that you want to check: It ain't over till it's over
Original string: It ain't over till it's over
Cleaned string: itaintovertillitsover
Reversed string: revostillitrevotniati
Your provided string is not a palindrome
/tmp/uaxwxHanYX.o
String that you want to check: radar
Original string: radar
Cleaned string: radar
Reversed string: radar
Your provided string is a palindrome.
/tmp/Bv2tV0IZlm.o
```

```
String that you want to check: When you come to a fork in the
road, take it
Original string: When you come to a fork in the road, take it
Cleaned string: whenyoucometoaforkintheroadtakeit
Reversed string: tiekatdaorehtnikrofaotemocuoynehw
Your provided string is not a palindrome
/tmp/YFDjUc0Ldm.o
String that you want to check: Marge lets Norah see Sharon's
telegram.
Original string: Marge lets Norah see Sharon's telegram.
Cleaned string: margeletsnorahseesharonstelegram
Reversed string: margeletsnorahseesharonstelegram
Your provided string is a palindrome.
```

Screenshot:

```
/tmp/GdLMGZnCc4.o
String that you want to check: Drab as a fool, aloof as a bard.
Original string: Drab as a fool, aloof as a bard.
Cleaned string: drabasafoolaloofasabard
Reversed string: drabasafoolaloofasabard
Your provided string is a palindrome.
```

```
/tmp/s9Q7YwBVYV.o
String that you want to check: It ain't over till it's over
Original string: It ain't over till it's over
Cleaned string: itaintovertillitsover
Reversed string: revostillitrevotniati
Your provided string is not a palindrome
```

```
/tmp/uaxwxHanYX.o
String that you want to check: radar
Original string: radar
Cleaned string: radar
Reversed string: radar
Your provided string is a palindrome.
```

```
/tmp/Bv2tV0IZlm.o
String that you want to check: When you come to a fork in the road, take it
Original string: When you come to a fork in the road, take it
Cleaned string: whenyoucometoaforkintheroadtakeit
Reversed string: tiekatdaorehtnikrofaotemocuoynehw
Your provided string is not a palindrome
```

```
/tmp/YFDjUc0Ldm.o
String that you want to check: Marge lets Norah see Sharon's telegram.
Original string: Marge lets Norah see Sharon's telegram.
Cleaned string: margeletsnorahseesharonstelegram
Reversed string: margeletsnorahseesharonstelegram
Your provided string is a palindrome.
```

Solution 2:

Algorithm:

1. Firstly the libraries are imported and the constants that are required to solve the problem are defined in the algorithm.
2. After this, the function Planc for calculation is defined here
3. Then in the main function the file in initialized and checked whether the system is able to access the file or not
4. Then calculations are performed based on the formula obtained in the lab manual
5. Then the data is uploaded to the file
6. After the GNUplot script is written that utilizes the data.txt file for displaying the graph in form of a PNG

Step 1: Writing the C program

Code:

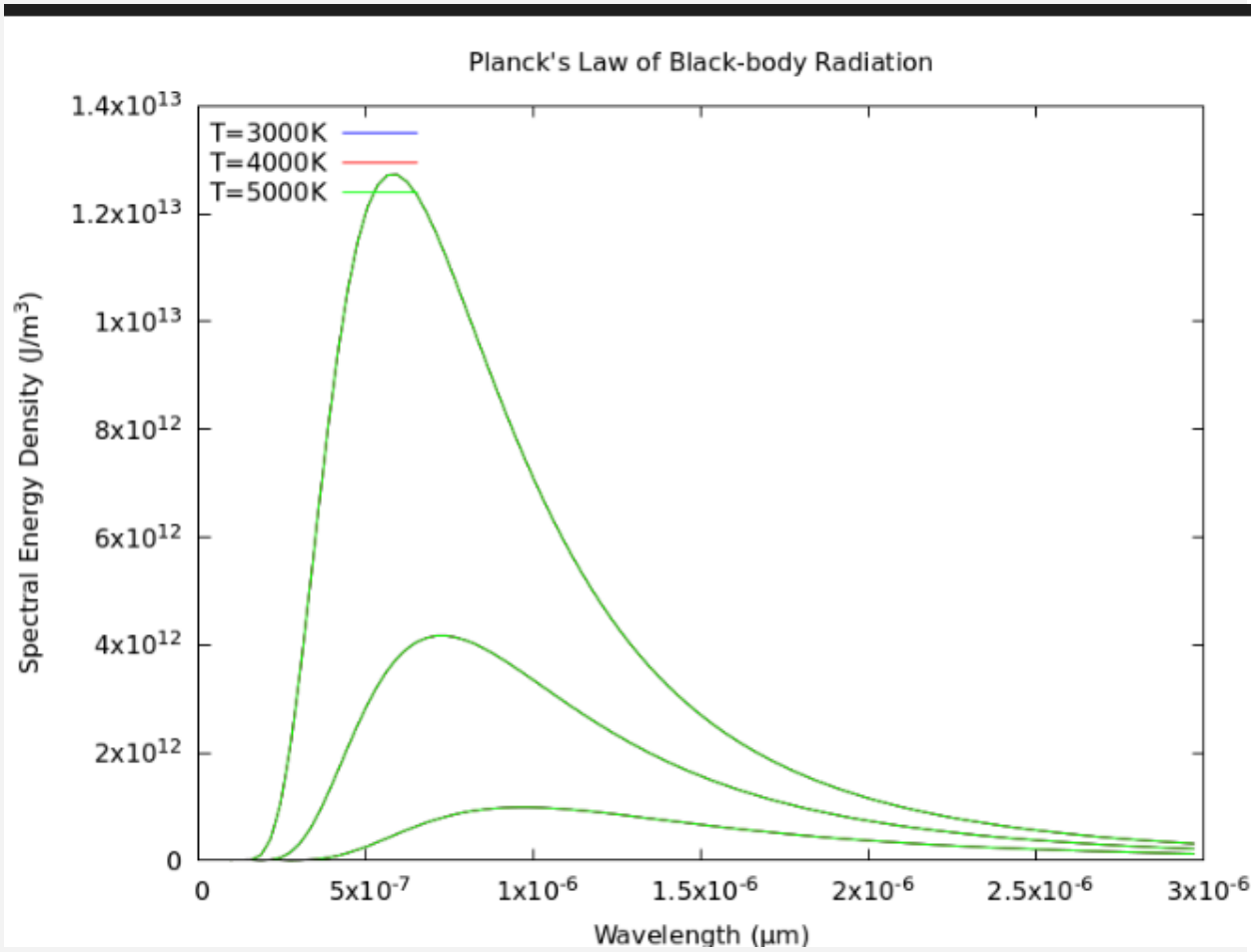
```c
#include <stdio.h>
#include <math.h>

#define C 3.0e8         // Speed of light (m/s)
#define H 6.626e-34     // Planck constant (J-s)
#define K 1.38e-23      // Boltzmann constant (J/K)
#define START_LAMBDA 0.1e-6  // Start wavelength (m)
#define END_LAMBDA 3.0e-6    // End wavelength (m)
#define NUM_POINTS 100   // Number of data points

double planck(double lambda, double T) {
    double exp_term = exp((H * C) / (lambda * K * T));
    return (2 * H * pow(C, 2)) / (pow(lambda, 5) * (exp_term - 1));
}

int main() {
    double lambda, temp;
    FILE *fp;

    fp = fopen("data.txt", "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        return 1;
    }

    for (temp = 3000; temp <= 5000; temp += 1000) {
        for (lambda = START_LAMBDA; lambda <= END_LAMBDA; lambda += (END_LAMBDA - START_LAMBDA) / NUM_POINTS) {
            fprintf(fp, "%e\t%e\t%e\n", lambda, temp, planck(lambda, temp));
        }
        fprintf(fp, "\n");
    }
```

Step 2: Writing the GNU Plot Script

Code:

```
set term pngcairo size 800,600 enhanced font 'Verdana,12'
set output 'blackbody_radiation.png'
set title "Planck's Law of Black-body Radiation"
set xlabel "Wavelength (μm)"
set ylabel "Spectral Energy Density (J/m^3)"
set key top left
plot "data.txt" using 1:3 title "T=3000K" with lines lc rgb "blue", \
     "" using 1:3 title "T=4000K" with lines lc rgb "red", \
     "" using 1:3 title "T=5000K" with lines lc rgb "green"
```

Output:



Data Output Screenshots:

1.000000e-07	3.000000e+03	1.675308e-02
1.290000e-07	3.000000e+03	2.285081e+02
1.580000e-07	3.000000e+03	7.682427e+04
1.870000e-07	3.000000e+03	3.683700e+06
2.160000e-07	3.000000e+03	5.627983e+07
2.450000e-07	3.000000e+03	4.163877e+08
2.740000e-07	3.000000e+03	1.894068e+09
3.030000e-07	3.000000e+03	6.128032e+09
3.320000e-07	3.000000e+03	1.548751e+10
3.610000e-07	3.000000e+03	3.256060e+10
3.900000e-07	3.000000e+03	5.948756e+10
4.190000e-07	3.000000e+03	9.744173e+10
4.480000e-07	3.000000e+03	1.464143e+11
4.770000e-07	3.000000e+03	2.052920e+11
5.060000e-07	3.000000e+03	2.721265e+11
5.350000e-07	3.000000e+03	3.444766e+11
5.640000e-07	3.000000e+03	4.197362e+11
5.930000e-07	3.000000e+03	4.953964e+11
6.220000e-07	3.000000e+03	5.692255e+11
6.510000e-07	3.000000e+03	6.393711e+11
6.800000e-07	3.000000e+03	7.043985e+11
7.090000e-07	3.000000e+03	7.632840e+11
7.380000e-07	3.000000e+03	8.153784e+11
7.670000e-07	3.000000e+03	8.603554e+11
7.960000e-07	3.000000e+03	8.981535e+11
8.250000e-07	3.000000e+03	9.289189e+11
8.540000e-07	3.000000e+03	9.529531e+11
8.830000e-07	3.000000e+03	9.706665e+11
9.120000e-07	3.000000e+03	9.825408e+11
9.410000e-07	3.000000e+03	9.890972e+11
9.700000e-07	3.000000e+03	9.908723e+11
9.990000e-07	3.000000e+03	9.883999e+11

1.115000e-06	3.000000e+03	9.459595e+11
1.144000e-06	3.000000e+03	9.294162e+11
1.173000e-06	3.000000e+03	9.112553e+11
1.202000e-06	3.000000e+03	8.917907e+11
1.231000e-06	3.000000e+03	8.713017e+11
1.260000e-06	3.000000e+03	8.500350e+11
1.289000e-06	3.000000e+03	8.282073e+11
1.318000e-06	3.000000e+03	8.060073e+11
1.347000e-06	3.000000e+03	7.835984e+11
1.376000e-06	3.000000e+03	7.611216e+11
1.405000e-06	3.000000e+03	7.386969e+11
1.434000e-06	3.000000e+03	7.164263e+11
1.463000e-06	3.000000e+03	6.943957e+11
1.492000e-06	3.000000e+03	6.726764e+11
1.521000e-06	3.000000e+03	6.513272e+11
1.550000e-06	3.000000e+03	6.303960e+11
1.579000e-06	3.000000e+03	6.099207e+11
1.608000e-06	3.000000e+03	5.899312e+11
1.637000e-06	3.000000e+03	5.704501e+11
1.666000e-06	3.000000e+03	5.514935e+11
1.695000e-06	3.000000e+03	5.330723e+11
1.724000e-06	3.000000e+03	5.151929e+11
1.753000e-06	3.000000e+03	4.978578e+11
1.782000e-06	3.000000e+03	4.810661e+11
1.811000e-06	3.000000e+03	4.648141e+11
1.840000e-06	3.000000e+03	4.490960e+11
1.869000e-06	3.000000e+03	4.339040e+11
1.898000e-06	3.000000e+03	4.192288e+11
1.927000e-06	3.000000e+03	4.050597e+11
1.956000e-06	3.000000e+03	3.913852e+11
1.985000e-06	3.000000e+03	3.781930e+11
2.014000e-06	3.000000e+03	3.654702e+11

2.101000e-06	3.000000e+03	3.299836e+11
2.130000e-06	3.000000e+03	3.190031e+11
2.159000e-06	3.000000e+03	3.084239e+11
2.188000e-06	3.000000e+03	2.982324e+11
2.217000e-06	3.000000e+03	2.884152e+11
2.246000e-06	3.000000e+03	2.789590e+11
2.275000e-06	3.000000e+03	2.698508e+11
2.304000e-06	3.000000e+03	2.610781e+11
2.333000e-06	3.000000e+03	2.526283e+11
2.362000e-06	3.000000e+03	2.444895e+11
2.391000e-06	3.000000e+03	2.366499e+11
2.420000e-06	3.000000e+03	2.290982e+11
2.449000e-06	3.000000e+03	2.218233e+11
2.478000e-06	3.000000e+03	2.148147e+11
2.507000e-06	3.000000e+03	2.080620e+11
2.536000e-06	3.000000e+03	2.015553e+11
2.565000e-06	3.000000e+03	1.952851e+11
2.594000e-06	3.000000e+03	1.892420e+11
2.623000e-06	3.000000e+03	1.834172e+11
2.652000e-06	3.000000e+03	1.778022e+11
2.681000e-06	3.000000e+03	1.723887e+11
2.710000e-06	3.000000e+03	1.671688e+11
2.739000e-06	3.000000e+03	1.621349e+11
2.768000e-06	3.000000e+03	1.572797e+11
2.797000e-06	3.000000e+03	1.525962e+11
2.826000e-06	3.000000e+03	1.480777e+11
2.855000e-06	3.000000e+03	1.437176e+11
2.884000e-06	3.000000e+03	1.395099e+11
2.913000e-06	3.000000e+03	1.354485e+11
2.942000e-06	3.000000e+03	1.315277e+11
2.971000e-06	3.000000e+03	1.277421e+11