Experiment 8

Student Name: Ishu	**UID: 22BCS15695**
Branch: BE-CSE	**Section/Group: 603_FL_IOT-B**
Semester: 5th	**Date of Performance:14/10/2024**
Subject Name: AP	**Subject Code: 22CSP-314**

1. **Aim:** Marc loves cupcakes, but he also likes to stay fit. Each cupcake has a calorie count, and Marc can walk a distance to expend those calories. If Marc has eaten j cupcakes so far, after eating a cupcake with c calories he must walk at least 2^j x c miles to maintain his weight.

2. **Objective:** The objective of this program is to determine the minimum distance Marc must walk to maintain his weight after consuming a specified number of cupcakes, each with a given calorie count. The challenge lies in the fact that Marc can eat the cupcakes in any order, and the goal is to find the optimal order that minimizes the total distance walked.

3. **Implementation/Code:**

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;

long marcsCakewalk(vector<int>& calorie) {
    sort(calorie.rbegin(), calorie.rend());
    long totalMiles = 0;

    for (size_t i = 0; i < calorie.size(); ++i) {
        totalMiles += calorie[i] * pow(2, i);
    }

    return totalMiles;
}
```

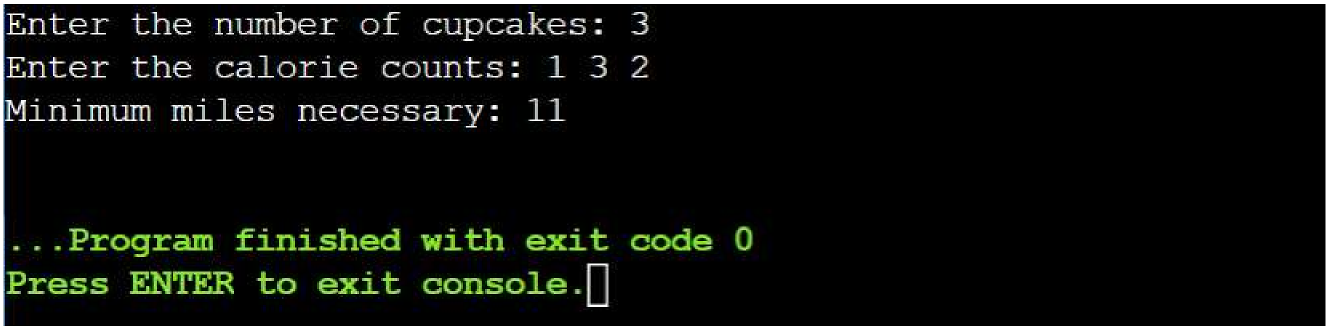
```
int main() {
    int n;
    cout << "Enter the number of cupcakes: ";
    cin >> n;

    vector<int> calorie(n);
    cout << "Enter the calorie counts: ";
    for (int i = 0; i < n; ++i) {
        cin >> calorie[i];
    }

    long result = marcsCakewalk(calorie);
    cout << "Minimum miles necessary: " << result << endl;

    return 0;
}
```

4. Output:



```
Enter the number of cupcakes: 3
Enter the calorie counts: 1 3 2
Minimum miles necessary: 11


...Program finished with exit code 0
Press ENTER to exit console.
```