CS 336 Assignment 1

January 26th 2025

Problem 1.

A group of n friends came to a restaurant, with the i'th person having $money[i]$ dollars. Looking at the menu, they see m dishes, with the j'th dish costing $price[j]$ dollars. For each friend, you want to find which dishes they can afford. In other words, for each friend i, you want to find the number of dishes j so that $price[j] \leq money[i]$.

Please implement the following function:

```
vector<int> count_dishes(const vector<int>& money, const vector<int>& prices)
```

The function accepts takes as input *money* – the amount of money each person has – and *prices* – the cost of each dish. The function must return the vector which has the same size as *money*; the i'th element of the returned vector should store the number of dishes the i'th person can afford. Implement this function, name the file **student_code_1.h** and submit on gradescope. You can find a template file attached to the assignment.

E.g. for $money = [5, 3, 4, 2, 4]$ and $prices = [2, 3, 5]$ (the first test), the output is $[3, 2, 2, 1, 2]$:

- The first person can buy all dishes.
- The second person can buy dishes with costs 2 and 3.
- The third person can buy dishes with costs 2 and 3.
- The fourth person can only buy a dish with cost 2.
- The fifth person can buy dishes with costs 2 and 3.

Time limit For every test, the function must take less than 1 second. You can assume that $1 \leq n, m \leq 2 \cdot 10^5$.

Scoring There are 100 small test cases ($n, m \leq 100$) and 100 large test cases. If your solution passes all of the small test cases you get 10 points. If it passes all of the large test cases you get 15 more points for a total of 25. There's no credit for solving some of the small or large test cases - scores for this problem will be either 0, 10, 15 or 25.

Problem 2.

Alice has chosen `n` holiday gifts for her friends. Alice knows that each gift will have a different price during Black Friday. The Black Friday's price of gift `i` will be `giftPrices[i].friday`. The price of item `i` before Black Friday will be `giftPrices[i].before`; and the price of gift `i` after Black Friday is going to be `giftPrices[i].after`.

Alice wants to plan when to buy each gift to minimize the total cost of all gifts. The challenge is that Alice can buy at most `k` gifts on Black Friday. She, however, can buy any number of gifts before and after Black Friday. Design and implement an algorithm for finding the minimum total cost of all `n` gifts. Implement the following function

```
long long int MinCost(const vector<Price>& giftPrices, int k)
```

The parameters of this function are `giftPrices` – the list of prices – and $k \geq 0$ – the maximum number of items Alice can buy on the Black Friday. The input vector `giftPrices` contains n elements and each element contains three fields: `friday`, `before`, and `after`, as described above. Function `MinCost` should return the minimum possible total cost of all gifts. Your file should be named **student_code_2** and submitted to gradescope. You can find a template file attached to the assignment. You can fill it out and submit to gradescope without changing the filename.

Time limit Your program should pass each tests in no more than 1 second. The number of gifts is between 1 and $2 \cdot 10^5$. Each price is between 1 and 10^9.

Scoring There are 100 small test cases ($n, k \leq 100$) and 100 large test cases. If your solution passes all of the small test cases you get 10 points. If it passes all of the large test cases you get 15 more points for a total of 25. There's no credit for solving some of the small or large test cases - scores for this problem will be either 0, 10, 15 or 25.