

SI@UCF Computer Science Camp
Intro to Competitive Programming in C++
Quiz #2
Date: 6/25/2025

Name: _____

1) (20 pts) Complete the program below so that it solves the following problem:

The first line of input contains two positive integers, n ($2 \leq n \leq 200,000$), and t ($3 \leq t \leq 2 \times 10^9$). The following n lines contain one positive integer each. Each of these integers is distinct and in between 1 and 10^9 . Output “Yes” if there exist two different integers on the list that add up to t exactly, and output, “No”, if no two integers in the list exist that add up to t exactly. To get significant credit, your solution must run in $O(n \lg n)$ time. (This means you can’t do a double for loop through the data.) **Do not use a set or map in your solution. (This will be follow up question.)**

```
using namespace std;
#include <bits/stdc++.h>

int main() {

    int n, t;
    cin >> n >> t;
    vector<int> vals(n);
    for (int i=0; i<n; i++)
        cin >> vals[i];

    /*** PUT SOLUTION HERE ***/

    return 0;
}
```

2) (15 pts) Alter the solution in question #1 so that instead of sorting the input values (this is what you should have done in the first step...) you avoid sorting the input values and use a set instead. In doing this, make sure you account for the fact that the two numbers that have to add up to the target have to be distinct. Write the code for this solution below. (Just the part that goes in the PUT SOLUTION HERE portion of the code on the previous page.)

3) (10 pts) Jerrod is writing a program that reads in votes for an election and needs to output the results sorted in order by number of votes, breaking ties by name. His idea is to use two maps, one `map<string,int>` and one `map<int,string>`. The first map would map each name to the number of votes. He would read in all the input data and build this map. Then, he would essentially copy that data into the second map, making the key the number of votes and the associated value the name. Unfortunately, he got a wrong answer when submitting his solution to this problem. Fundamentally, what is his error?

4) (25 pts) In the game of minesweeper, the game board is a 2D grid where each square is either an underscore ('_') or a star (*). The stars represent bombs and the underscores represent valid squares. We define a safe valid square to be a valid square that has 2 or fewer adjacent bombs. A bomb is adjacent to a square if it is next to it either up, down, left, right or in one of the four possible diagonal directions. (Thus, the maximum number of bombs that could be adjacent to a valid square is 8.) Write a function that takes in the game board for minesweeper (a vector of strings, where all characters are either '_' or '*'), and returns the number of the valid squares on that grid that are safe. **Please define your own DX/DY arrays before the function.**

```
const vector<int> DX = {                                                                                                                                            };
```

```
const vector<int> DY = {                                                                                                                                            };
```

```
int numSafeValid(vector<string> grid) {
```

```
}
```

```
using namespace std;
#include <bits/stdc++.h>

// Indexes of each row, column and diagonal of a 3 x 3 matrix
// stored in a 1D vector.
const vector<vector<int>> LISTS = {{0,1,2}, {3,4,5}, {6,7,8}, {0,3,6},
                                   {1,4,7}, {2,5,8}, {0,4,8}, {2,4,6}};

const int S = 3;
const int N = S*S;
void go(vector<int>& perm, vector<bool>& used, int k);
bool valid(vector<int>& perm);
void print(vector<int>& perm);

int main() {
    vector<int> perm(N);
    vector<bool> used(N, false);
    go(perm, used, 0);
    return 0;
}

void go(vector<int>& perm, vector<bool>& used, int k) {

    if (k == perm.size()) {
        if (valid(perm))
            print(perm);
        return;
    }
    /** FILL IN CODE HERE ***/
    for (int i=0; i<perm.size(); i++) {

        }
```

```

// Returns true iff this permutation is a valid 3 x 3 Magic Square
// Make use of LISTS!!! Returns false otherwise.
bool valid(vector<int>& perm) {

    /*** FILL IN THIS ENTIRE FUNCTION ***/

}

void print(vector<int>& perm) {
    for (int i=0; i<S; i++) {
        for (int j=0; j<S; j++)
            cout << perm[S*i+j]+1 << " ";
        cout << endl;
    }
    cout << endl;
}

```

6) (5 pts) What animal appears on the Panda Express logo?

Vector Class and Algorithm Documentation

```

// Adds val to the end of this vector.
void push_back (const value_type& val);

// Returns the size of this vector.
int size();

// Sorts vector v in increasing order.
sort(v.begin(), v.end())

```

Scratch Page – Please clearly mark any work on this page you would like graded.