Name = Harsh Raj Chouadhary

UID = 22BCS11231

Problem 1

major.cpp > findMajorityElement(const vector<int>&)

```cpp
#include <iostream>
#include <vector>

using namespace std;

int findMajorityElement(const vector<int>& nums) {
    int count = 0;
    int candidate = 0;

    for (int num : nums) {
        if (count == 0) {
            candidate = num;
        }
        count += (num == candidate) ? 1 : -1;
    }

    count = 0;
    for (int num : nums) {
        if (num == candidate) {
            count++;
        }
    }

    return candidate;
}

int main() {
    int n;
    cout << "Enter the number of elements in the array: ";
    cin >> n;

    vector<int> nums(n);
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; ++i) {
        cin >> nums[i];
    }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\ISHAM\OneDrive\Documents\coding> cd "c:
Enter the number of elements in the array: 5
Enter the elements of the array: 3 2 4 2 4
The majority element is: 4
PS C:\Users\ISHAM\OneDrive\Documents\coding>
```

Problem 2

major.cpp | shape.cpp | container.cpp | happymods.cpp | minimumjobs.cpp

shape.cpp > generatePascalsTriangle(int)

```cpp
#include <iostream>
#include <vector>

using namespace std;

vector<vector<int>> generatePascalsTriangle(int numRows) {
    vector<vector<int>> triangle;

    for (int i = 0; i < numRows; ++i) {
        vector<int> row(i + 1, 1);

        for (int j = 1; j < i; ++j) {
            row[j] = triangle[i - 1][j - 1] + triangle[i - 1][j];
        }

        triangle.push_back(row);
    }

    return triangle;
}

int main() {
    int numRows;
    cout << "Enter the number of rows for Pascal's Triangle: ";
    cin >> numRows;

    vector<vector<int>> pascalsTriangle = generatePascalsTriangle(numRows);

    cout << "Pascal's Triangle with " << numRows << " rows:" << endl;
    for (const auto& row : pascalsTriangle) {
        for (int num : row) {
            cout << num << " ";
        }
        cout << endl;
    }

    return 0;
}
```

```
Enter the number of rows for Pascal's Triangle: 4
Pascal's Triangle with 4 rows:
1
1 1
1 2 1
1 3 3 1
PS C:\Users\ISHAM\OneDrive\Documents\coding>
```

Problem 3

container.cpp > maxArea(const vector<int>&)

```cpp
#include <bits/stdc++.h>

using namespace std;

int maxArea(const vector<int>& height) {
    int maxWater = 0;
    int left = 0, right = height.size() - 1;

    while (left < right) {
        int width = right - left;
        int currentHeight = min(height[left], height[right]);
        maxWater = max(maxWater, width * currentHeight);

        if (height[left] < height[right]) {
            ++left;
        } else {
            --right;
        }
    }

    return maxWater;
}

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

    vector<int> height(n);
    cout << "Enter the heights of the lines: ";
    for (int i = 0; i < n; ++i) {
        cin >> height[i];
    }

    int result = maxArea(height);
    cout << "The maximum amount of water a container can store is: " << result << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ISHAM\OneDrive\Documents\coding> cd "c:\Users\ISHAM\OneDrive\Documents\coding\"
Enter the number of lines: 3
Enter the heights of the lines: 2
3
4
The maximum amount of water a container can store is: 4
PS C:\Users\ISHAM\OneDrive\Documents\coding>
```

Problem 4

major.cpp | shape.cpp | container.cpp | happymods.cpp | minimumjobs.cpp

happymods.cpp > maxHappyGroups(int, vector<int>&)

```cpp
#include <bits/stdc++.h>

using namespace std;

int maxHappyGroups(int batchSize, vector<int>& groups) {
    vector<int> remainderCount(batchSize, 0);
    int happyGroups = 0;

    for (int group : groups) {
        int remainder = group % batchSize;
        if (remainder == 0) {
            happyGroups++;
        } else if (remainderCount[batchSize - remainder] > 0) {
            happyGroups++;
            remainderCount[batchSize - remainder]--;
        } else {
            remainderCount[remainder]++;
        }
    }

    int remainingGroups = 0;
    for (int i = 1; i < batchSize; ++i) {
        remainingGroups += remainderCount[i];
    }

    happyGroups += remainingGroups / batchSize;

    return happyGroups;
}

int main() {
    int batchSize, n;
    cout << "Enter the batch size: ";
    cin >> batchSize;

    cout << "Enter the number of groups: ";
    cin >> n;

    vector<int> groups(n);
    cout << "Enter the size of each group: ";
    for (int i = 0; i < n; ++i) {
        cin >> groups[i];
    }

    int result = maxHappyGroups(batchSize, groups);
    cout << "The maximum number of happy groups is: " << result << endl;

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

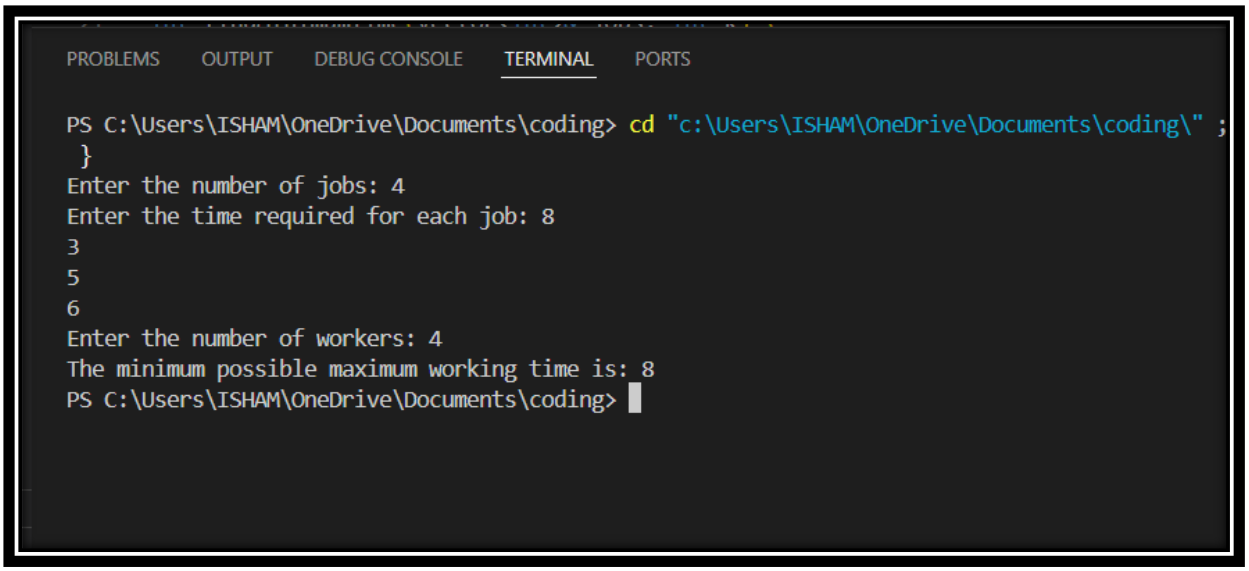
```
PS C:\Users\ISHAM\OneDrive\Documents\coding> cd "c:\Users\ISHAM\OneDrive\Documents\coding\" ; j
Enter the batch size: 4
Enter the number of groups: 4
Enter the size of each group: 3 3 2 4
The maximum number of happy groups is: 1
PS C:\Users\ISHAM\OneDrive\Documents\coding>
```

Problem 5

major.cpp | shape.cpp | container.cpp | happymods.cpp | minimumjobs.cpp

minimumjobs.cpp > isFeasible(vector<int>&, vector<int>&, int, int)

```cpp
#include <bits/stdc++.h>

using namespace std;

bool isFeasible(vector<int>& jobs, vector<int>& workers, int idx, int maxTime) {
    if (idx == jobs.size()) return true;

    for (int i = 0; i < workers.size(); ++i) {
        if (workers[i] + jobs[idx] <= maxTime) {
            workers[i] += jobs[idx];
            if (isFeasible(jobs, workers, idx + 1, maxTime)) return true;
            workers[i] -= jobs[idx];
        }

        if (workers[i] == 0) break;
    }

    return false;
}

int findMinimumTime(vector<int>& jobs, int k) {
    sort(jobs.rbegin(), jobs.rend());

    int left = jobs[0], right = accumulate(jobs.begin(), jobs.end(), 0), result = right;

    while (left <= right) {
        int mid = left + (right - left) / 2;
        vector<int> workers(k, 0);

        if (isFeasible(jobs, workers, 0, mid)) {
            result = mid;
            right = mid - 1;
        } else {
            left = mid + 1;
        }
    }

    return result;
}

int main() {
    int n, k;
    cout << "Enter the number of jobs: ";
    cin >> n;

    vector<int> jobs(n);
    cout << "Enter the time required for each job: ";
    for (int i = 0; i < n; ++i) {
        cin >> jobs[i];
    }

    cout << "Enter the number of workers: ";
    cin >> k;

    int result = findMinimumTime(jobs, k);
    cout << "The minimum possible maximum working time is: " << result << endl;

    return 0;
}

```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\ISHAM\OneDrive\Documents\coding> cd "c:\Users\ISHAM\OneDrive\Documents\coding\" ;
}
Enter the number of jobs: 4
Enter the time required for each job: 8
3
5
6
Enter the number of workers: 4
The minimum possible maximum working time is: 8
PS C:\Users\ISHAM\OneDrive\Documents\coding>