TASK 1:

#include<iostream>

#include<vector>

using namespace std;

int main()

{

vector<int>baha;

for(int i=0;i<11;i++){

baha.push\_back(i);

}

int x=baha.size();

for(int i=0;i<x;i++){

cout<<baha[i]<<endl;

}

baha.push\_back(5);

baha.erase(baha.begin()+5);

x=baha.size();

for(int i=0;i<x;i++){

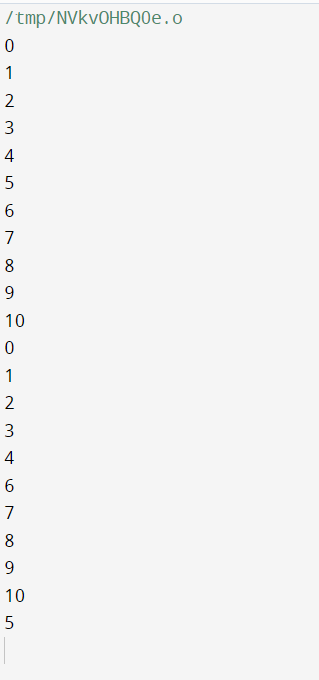
cout<<baha[i]<<endl;

}

return 0;

}

OUTPUT:



TASK 2:

#include <iostream>

#include <vector>

using namespace std;

double calculateMean(const vector<int>& grades) {

int sum = 0;

for (int grade : grades) {

sum += grade;

}

return static\_cast<double>(sum) / grades.size();

}

double calculateMedian(vector<int>& grades) {

// Bubble sort

int n = grades.size();

for (int i = 0; i < n - 1; ++i) {

for (int j = 0; j < n - i - 1; ++j) {

if (grades[j] > grades[j + 1]) {

// Swap

int temp = grades[j];

grades[j] = grades[j + 1];

grades[j + 1] = temp;

}

}

}

if (n % 2 == 0) {

int mid1 = grades[n / 2 - 1];

int mid2 = grades[n / 2];

return (static\_cast<double>(mid1 + mid2)) / 2;

} else {

return grades[n / 2];

}

}

int calculateMode(const vector<int>& grades) {

vector<int> freq(101, 0); // Assuming grades are between 0 and 100

for (int grade : grades) {

freq[grade]++;

}

int mode = 0, maxFreq = 0;

for (int i = 0; i < freq.size(); ++i) {

if (freq[i] > maxFreq) {

maxFreq = freq[i];

mode = i;

}

}

return mode;

}

int main() {

int numPairs;

cout << "Enter the number of name/grade pairs: ";

cin >> numPairs;

vector<string> names;

vector<int> grades;

for (int i = 0; i < numPairs; ++i) {

string name;

int grade;

cout << "Enter name #" << i + 1 << ": ";

cin >> name;

cout << "Enter grade #" << i + 1 << ": ";

cin >> grade;

names.push\_back(name);

grades.push\_back(grade);

}

double mean = calculateMean(grades);

double median = calculateMedian(grades);

int mode = calculateMode(grades);

cout << "Mean of the grades: " << mean << endl;

cout << "Median of the grades: " << median << endl;

cout << "Mode of the grades: " << mode << endl;

cout << "Names of students with the mode grade (" << mode << "):" << endl;

for (size\_t i = 0; i < grades.size(); ++i) {

if (grades[i] == mode) {

cout << names[i] << endl;

}

}

return 0;

}

OUTPUT:

