#include <iostream>

#include <iomanip>

#include <vector>

#include <string>

using namespace std;

const int MIN\_SENIORS = 10;

const int MAX\_SENIORS = 36;

const double CARER\_COST = 0.0;

const double SENIOR\_COSTS[][3] = {

{150, 14.00, 21.00}, // 12-16 seniors

{190, 13.50, 20.00}, // 17-26 seniors

{225, 13.00, 19.00} // 27-39 seniors

};

double calculateCost(int numSeniors) {

int rangeIndex;

if (numSeniors >= 12 && numSeniors <= 16)

rangeIndex = 0;

else if (numSeniors >= 17 && numSeniors <= 26)

rangeIndex = 1;

else if (numSeniors >= 27 && numSeniors <= 39)

rangeIndex = 2;

else {

cout << "Invalid number of seniors." << endl;

return -1.0;

}

double cost = SENIOR\_COSTS[rangeIndex][0];

return cost;

}

bool task1(int& numSeniors) {

while (true) {

cout << "Enter the number of senior citizens interested in the outing: ";

cin >> numSeniors;

if (numSeniors < MIN\_SENIORS || numSeniors > MAX\_SENIORS) {

cout << "Outing cannot proceed with this number of seniors." << endl;

continue;

}

return true;

}

}

struct Participant {

string name;

double amountPaid;

};

void task2(int numParticipants, double costPerPerson, int numSeniors) {

double amountPaid;

cout << "Enter the amount paid by each participant: $";

cin >> amountPaid;

vector<Participant> participants;

string name;

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

cout << "Enter participant #" << (i + 1) << "'s name: ";

cin >> name;

participants.push\_back({name, amountPaid});

}

int numCarers = (numSeniors > 24) ? 3 : 2; // Additional carer needed if more than 24 seniors

double totalCollected = numParticipants \* amountPaid;

totalCollected += (numSeniors + numCarers) \* costPerPerson; // Adding cost for carers and additional participants

cout << "\nList of participants:" << endl;

cout << "-----------------------------" << endl;

for (const auto& participant : participants) {

cout << participant.name << ": $" << participant.amountPaid << endl;

}

cout << "-----------------------------" << endl;

cout << "Total amount collected: $" << totalCollected << endl;

cout << "Number of caretakers: " << numCarers << endl; // Output the number of caretakers

}

void task3(double totalCost, double totalCollected) {

if (totalCollected >= totalCost) {

cout << "The outing has broken even." << endl;

} else {

cout << "The outing has made a profit of $" << totalCost - totalCollected << "." << endl;

}

}

int main() {

int numSeniors;

if (!task1(numSeniors))

return 0;

double totalCost = calculateCost(numSeniors);

double costPerPerson = totalCost / (numSeniors + 2); // Additional 2 for caretakers

cout << "Total cost of the outing: $" << fixed << setprecision(2) << totalCost << endl;

cout << "Cost per person: $" << fixed << setprecision(2) << costPerPerson << endl;

int numParticipants;

cout << "Enter the number of participants going on the outing: ";

cin >> numParticipants;

task2(numParticipants, costPerPerson, numSeniors);

task3(totalCost, totalCost); // Assuming the total collected is equal to the total cost for simplicity

return 0;

}