**/\*Write a C++ program that implements the following functions:**

**a) calculateTotalExpenses: Calculates and returns the total sum of expenses across all categories.**

**b) findMaxExpense: Finds and returns the maximum expense among all categories.**

**c) findMinExpense: Finds and returns the minimum expense among all categories.**

**d) averageExpense: Calculates and returns the average expense per category.**

**e) updateExpense: Updates the expense for a specific category.**

**f) addExpense: Inserts expenses for a particular month.**

**g) removeExpense: Deletes expenses for a particular month.**

**h) printExpenses: Prints all expenses in each category.\*/**

**#include <iostream>**

**using namespace std;**

**class one{**

**string categories[10];**

**int expenses[10];**

**public:**

**void cat(int n) {**

**cout << "Enter the names of the categories" << endl;**

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

**cout << i + 1 << ". ";**

**cin >> categories[i];**

**expenses[i] = 0;**

**}**

**}**

**int catnum() {**

**int catnumber;**

**cout << "Enter the category number: ";**

**cin >> catnumber;**

**return catnumber - 1;**

**}**

**void addExpense() {**

**int catnumber = catnum();**

**cout << "Enter expense for category " << categories[catnumber] << ": ";**

**cin >> expenses[catnumber];**

**cout << "Expense added successfully!" << endl;**

**}**

**void removeExpense() {**

**int catnumber = catnum();**

**expenses[catnumber] = 0;**

**cout << "Expense removed successfully!" << endl;**

**}**

**void updateExpense() {**

**int catnumber = catnum();**

**cout << "Enter updated expense for category " << categories[catnumber] << ": ";**

**cin >> expenses[catnumber];**

**cout << "Expense updated successfully!" << endl;**

**}**

**void printExpenses() {**

**cout << "Expenses:" << endl;**

**for (int i = 0; i < 10; i++) {**

**if (!categories[i].empty()) {**

**cout << categories[i] << ": " << expenses[i] << endl;**

**}**

**}**

**}**

**int calculateTotalExpenses() {**

**int total = 0;**

**for (int i = 0; i < 10; i++) {**

**total += expenses[i];**

**}**

**return total;**

**}**

**int findMaxExpense() {**

**int maxExpense = expenses[0];**

**for (int i = 1; i < 10; i++) {**

**if (expenses[i] > maxExpense) {**

**maxExpense = expenses[i];**

**}**

**}**

**return maxExpense;**

**}**

**int findMinExpense() {**

**int minExpense = expenses[0];**

**for (int i = 1; i < 10; i++) {**

**if (expenses[i] < minExpense) {**

**minExpense = expenses[i];**

**}**

**}**

**return minExpense;**

**}**

**double averageExpense(int n) {**

**int totalExpenses = calculateTotalExpenses();**

**return static\_cast<double>(totalExpenses) / n;**

**}**

**};**

**int main() {**

**one x1;**

**cout<<"enter the number of categories "<<endl;**

**int n;**

**cin>>n;**

**x1.cat(n);**

**char choice;**

**do {**

**cout << "Choose among these "<<endl;**

**cout << "1. Add Expense"<<endl;**

**cout << "2. Remove Expense"<<endl;**

**cout << "3. Update Expense"<<endl;**

**cout << "4. Print Expenses"<<endl;**

**cout << "5. Calculate Total Expenses"<<endl;**

**cout << "6. Find Maximum Expense"<<endl;**

**cout << "7. Find Minimum Expense"<<endl;**

**cout << "8. Calculate Average Expense"<<endl;**

**cout << "9. Exit"<<endl;**

**cout << "Enter your choice: "<<endl;**

**cin >> choice;**

**switch (choice) {**

**case '1':**

**x1.addExpense();**

**break;**

**case '2':**

**x1.removeExpense();**

**break;**

**case '3':**

**x1.updateExpense();**

**break;**

**case '4':**

**x1.printExpenses();**

**break;**

**case '5':**

**cout << "Total Expenses: " << x1.calculateTotalExpenses() << endl;**

**break;**

**case '6':**

**cout << "Maximum Expense: " << x1.findMaxExpense() << endl;**

**break;**

**case '7':**

**cout << "Minimum Expense: " << x1.findMinExpense() << endl;**

**break;**

**case '8':**

**cout << "Average Expense: " << x1.averageExpense(n) << endl;**

**break;**

**case '9':**

**cout << "Exiting program." << endl;**

**break;**

**default:**

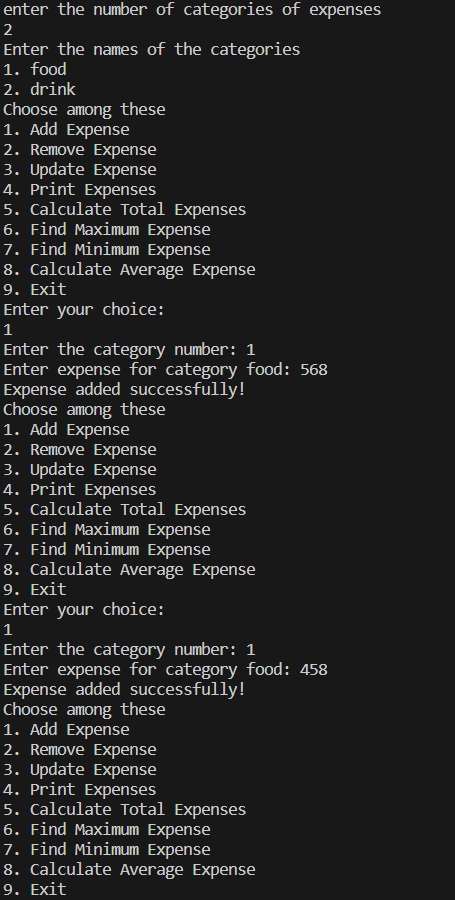
**cout << "Invalid choice. Please try again." << endl;**

**break;**

**}**

**} while (choice != '9');**

**}**

****