```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NAME_SIZE 35
#define POSITION_SIZE 50
#define PERSON_LIST_LENGTH 100
#define DESCRIPTION_SIZE 50
#define EXPENSE_LIST_LENGTH 100


//Structures
typedef struct
{
    int ID;
    char name[NAME_SIZE];
    char position[POSITION_SIZE];
    double salary;
} Employee;

typedef struct
{
    Employee array[PERSON_LIST_LENGTH];
    int length;
    int count;
} EmployeeList;

typedef struct
{
    int employee_id;
    char description[DESCRIPTION_SIZE];
    double cost;
}Expense;

typedef struct
{
   Expense array[EXPENSE_LIST_LENGTH];
   int length;
   int count;
}ExpenseList;



//Prototypes
EmployeeList create_employee_list();
int read_employees(EmployeeList *plistOfEmployees);
void print_employee(Employee employeeToPrint);
void print_employee_list(EmployeeList employeeListToPrint);
ExpenseList create_expense_list();
int read_expenses(ExpenseList *plistOfExpenses);
void print_expense(Expense expenseToPrint);
void print_expense_list(ExpenseList expenseListToPrint);
void calc_expenses(EmployeeList employee, ExpenseList expense);

//Main
int main()
{
    EmployeeList listOfEmployees = create_employee_list();
    int employeeCount = read_employees(&listOfEmployees);
```

```
    print_employee_list(listOfEmployees);


    ExpenseList listOfExpenses = create_expense_list();
    int expenseCount = read_expenses(&listOfExpenses);
    print_expense_list(listOfExpenses);

    calc_expenses(listOfEmployees, listOfExpenses);

return 0;
}


//Functions
EmployeeList create_employee_list()
{
    EmployeeList listOfEmployees;
    listOfEmployees.length = PERSON_LIST_LENGTH;
    listOfEmployees.count = 0;

   return listOfEmployees;
}

int read_employees(EmployeeList *plistOfEmployees)
{
    char employeeInformation[6][255] = {"1|Archer, Malory|CEO|500000.0",
                                        "2|Archer, Sterling|Agent|250000.0",
                                        "3|Kane, Lana|Agent|300000.0",
                                        "4|Figus, Ceril|Accountant|100000.0",
                                        "5|Tunt, Cheryl|Secretary|65000.0",
                                        "6|Poovey, Pam|HR|85000.0"
                                        };


 for (int i = 0; i < 6; i++)
 {
    Employee employeeToRead;
    int tokenRead = 0;
    char *token = strtok(employeeInformation[i], "|");
    while (token != NULL)
    {
       ++tokenRead;
       switch (tokenRead)
       {
       case 1:
           employeeToRead.ID = atoi(token);
           break;

       case 2:
            strcpy(employeeToRead.name, token);
            break;

        case 3:
            strcpy(employeeToRead.position, token);
            break;

        case 4:
            employeeToRead.salary = atof(token);
            break;
```

```
       default:
           break;
       }
       token = strtok(NULL, "|");
    }

    plistOfEmployees -> array[plistOfEmployees->count] = employeeToRead;
    plistOfEmployees -> count++;

 }

  return plistOfEmployees -> count;
}

void print_employee(Employee employeeToPrint)
{
    printf("\nID:       %d", employeeToPrint.ID);
    printf("\nName:     %s", employeeToPrint.name);
    printf("\nPosition: %s", employeeToPrint.position);
    printf("\nSalary:   %5.2f", employeeToPrint.salary);
}

void print_employee_list(EmployeeList employeeListToPrint)
{
    printf("\nI.S.I.S Employees\n");
    for (int i = 0; i < employeeListToPrint.count; i++)
    {
       print_employee(employeeListToPrint.array[i]);
       printf("\n");
    }

}

ExpenseList create_expense_list()
{
    ExpenseList listOfExpenses;
    listOfExpenses.length = EXPENSE_LIST_LENGTH;
    listOfExpenses.count = 0;

return listOfExpenses;
}

int read_expenses(ExpenseList *plistOfExpenses)
{
    char employeeExpenses[15][255] = {"1,Dinner,456.23",
                                      "1,Air Travel,692.12",
                                      "1,Spa Day,725.00",
                                      "2,Bar Bill,392.34",
                                      "2,Glenghoulie Blue,320.00",
                                      "2,Bar Bill,523.54",
                                      "3,Explosives,550.00",
                                      "3,Firearms,2343.56",
                                      "3,Amunition,245.98",
                                      "4,Lunch,52.45",
                                      "4,Ledger,23.45",
                                      "5,Copy Paper,56.23",
                                      "5,Staples,12.09",
                                      "6,Milk,4.50",
```

```
                                        "6,Cheese,7.89"};

    for (int i = 0; i < 15; i++)
    {
        Expense expenseToRead;
        int tokenRead = 0;
        char *token = strtok(employeeExpenses[i], ",");
        while(token != NULL)
        {
            ++tokenRead;
            switch (tokenRead)
            {
            case 1:
                expenseToRead.employee_id = atoi(token);
                break;

            case 2:
                    strcpy(expenseToRead.description, token);
                    break;

                case 3:
                    expenseToRead.cost = atof(token);
                    break;

            default:
                break;
            }
            token = strtok(NULL, ",");
        }
        plistOfExpenses -> array[plistOfExpenses -> count] = expenseToRead;
        plistOfExpenses -> count++;
    }
    return plistOfExpenses -> count;
}

void print_expense(Expense expenseToPrint)
{
    printf("\nEmployee ID: %d", expenseToPrint.employee_id);
    printf("\nDescription: %s", expenseToPrint.description);
    printf("\nCost:        %5.2f", expenseToPrint.cost);
}

void print_expense_list(ExpenseList expenseListToPrint)
{
    printf("\nI.S.I.S Expenses\n");
    for (int i = 0; i < expenseListToPrint.count; i++)
    {
       print_expense(expenseListToPrint.array[i]);
       printf("\n");
    }

}

void calc_expenses(EmployeeList employee, ExpenseList expense)
{
    Employee empCompare;
    Expense expCompare;
    double totalExpense = 0.0;
```

```
    for (int i = 0; i < employee.count; i++)
    {
        totalExpense = 0.0;
        empCompare = employee.array[i];
       printf("\n %d %s %s %f", empCompare.ID, empCompare.name,
empCompare.position, empCompare.salary);
        for (int j = 0; j < expense.count; j++)
        {
            expCompare = expense.array[j];
            if (empCompare.ID == expCompare.employee_id)
            {
                printf("\n        %s $%5.2f", expCompare.description,
expCompare.cost);
                totalExpense += expCompare.cost;
            }

        }
         printf("\nTotal: %5.2f\n", totalExpense);
    }
}
```