Data Name

# Problem statement:

# A family wants to manage their monthly expenses. In order to complete this task, the family needs an

# application to store, for a given month, all their expenses. Each expense will be stored in the

# application using the following elements: day (of the month in which it was made, between 1 and 30),

# amount of money (positive integer) and expense type (one of: housekeeping, food, transport,

# clothing, internet, others).

# Feature list

|  |
| --- |
| **Feature** |
| 1. Add to the current day an expense. |
| 1. Insert to a given day an expense |
| 1. Remove all the expenses for a given day |
| 1. Remove all the expenses between 2 given days |
| 1. Remove all the expenses for a category from the current month. |
| 1. Write the entire list of expenses |
| 1. Write all the expenses for a category |
| 1. Writes all expenses for a given category with an amount of money < = > than a given value |
| 1. Write the total expense for a given category |
| 1. Write the day with the maximum expenses. |
| 1. Write the total daily expenses in ascending order by amount of money spent |
| 1. Write the daily expenses for a given category in ascending order by amount of money spent |
| 1. Filter the expenses, keeping only expenses in a given category |
| 1. Filter the expenses, keeping only expenses in a given category an amount of money < = > than a given value |
| 1. Undo the last operation (user may repeat this operation). |

## Running scenario 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User** | **Program** | **Description** |
| a |  |  | Ask for a command |
| b | Add today,13,net |  | Add to the current day an expense of 13 RON for category net |
| c |  |  | Ask for a command |
| d | add 25,100 food |  | insert to day 25 an expense of 100 RON for food. |
| e |  |  | Ask for a command |
| f | remove 15 |  | remove all the expenses for day 15. |
| g |  |  | Ask for a command |
| h | remove 2 to 9 |  | remove all the expenses between day 2 and day 9 |
| i |  |  | Ask for a command |
| j | remove food |  | remove all the expenses for food from the current month |
| k |  |  | Ask for a command |
| l | list |  | write the entire list of expenses |
| m |  |  | Ask for a command |
| n | list food |  | write all the expenses for food. |
| o |  |  | Ask for a command |
| p | list food > 5 |  | writes all expenses for food with an amount of money > 5 |
| q |  |  | Ask for a command |
| r | list internet = 44 |  | writes all expenses for internet with an amount of money = 44 |
| s |  |  | Ask for a command |
|  | sum |  | Writes the sum of money spent on category food |
|  |  |  | Ask for a command |
|  | max |  | Writes the day with the maximum expenses |
|  |  |  | Ask for a command |
|  | Sort 16 |  | Writes the expenses on day 16,sorted by the amount of money spent |
|  |  |  | Ask for a command |
|  | Sort food |  | Writes the expenses on food,sorted by the amount of money spent |
|  |  |  | Ask for a command |
|  | Filter food |  | –keep only expenses in category food |
|  |  |  | Ask for a command |
|  | filter books < 100 |  | – keep only expenses in category books with amount of money < 100 |
|  |  |  | Ask for a command |
|  | undo |  | Undo the last operation |
|  |  |  | Ask for a command |
| t | help |  | Show the menu |
| v |  |  | Ask for a command |
| u | exit |  | Exit the application |

## Tasks

|  |  |
| --- | --- |
| Id | Description |
| T1 | Add a new expense into the list |
| T2 | Modify expenses from the list. |
| T3 | Write the expenses having different properties |
| T4 | Implement user interface |

1. **Test case table for function findDay(expenseList,day)**

|  |  |
| --- | --- |
| **Data: expenseList,day** | **Result: findDay(expenseList,day)** |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  14 | 2 |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  2 | 5 |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  3 | -1 |

1. **Test case table for function findDayToDay(expenseList,day1,day2)**

|  |  |
| --- | --- |
| **Data: expenseList,day** | **Result: findDay(expenseList,day)** |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  14,18 | 1 |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  14,18 | 2 |
| (1, 20, 'food')  (15, 230, 'transport')  (14, 90, 'food')  (6, 520, 'others')  (2, 450, 'rent')  3,5 | -1 |