

Subject: PRF192 - Programming Fundamental with C

Workshop 4

Objectives:

- Enhance skills in modular programming by implementing distinct functions for ascending and descending sorting.
- Practice conditional logic to filter values based on a sales target.
- Develop real-world problem-solving skills with structured data handling in C.

Problem Description:

In a retail store, sales data for a specific period (e.g., a week or a month) needs to be analyzed. Managers require tools to organize and evaluate this data. The program must include functionalities to:

1. **Input:** Enter daily sales data for a given period (maximum 100 days).
2. **Display:** Show all sales data to analyze trends.
3. **Sort in Ascending Order:** Arrange sales data in increasing order and display it.
4. **Sort in Descending Order:** Arrange sales data in decreasing order and display it.
5. **Search for Values Greater Than a Given Sales Target:** Identify and list all sales values that exceed the specified target.

This extended problem involves writing modular functions to handle ascending and descending sorting, along with advanced search logic. These operations will provide better insights into sales performance.

Situation Description:

Retail managers often need to review sales trends to understand performance metrics. Sorting sales data in both ascending and descending order can help identify top-performing days or periods with low sales. Searching for sales values greater than a target allows them to gauge the effectiveness of promotions or peak sales periods.

Syntax Use in the Problem:

1. **Structs:**
 - Encapsulate the sales data and metadata for logical grouping.
2. **Functions:**
 - Input, display, ascending sorting, descending sorting, and advanced search functionalities are modularized.

3. **Loops:**

- For loops are used for sorting, searching, and data traversal.

4. **Conditional Statements:**

- Used in the search functionality to filter values based on the target.

Specific Requirements:

1. **Struct Definition:**

- Encapsulates:
 - An array of integers representing daily sales.
 - The number of days (size of the array).

2. **Functionalities:**

- Input: Collect daily sales figures.
- Display: Show all sales data with day labels.
- Sort in Ascending Order: Arrange and print sales figures in increasing order.
- Sort in Descending Order: Arrange and print sales figures in decreasing order.
- Search for Values Greater Than a Target: Identify and display all sales values exceeding the target.

3. **Constraints:**

- Maximum array size: 100 days.
- The number of days must be validated (1–100).

Hint: Code Design

```

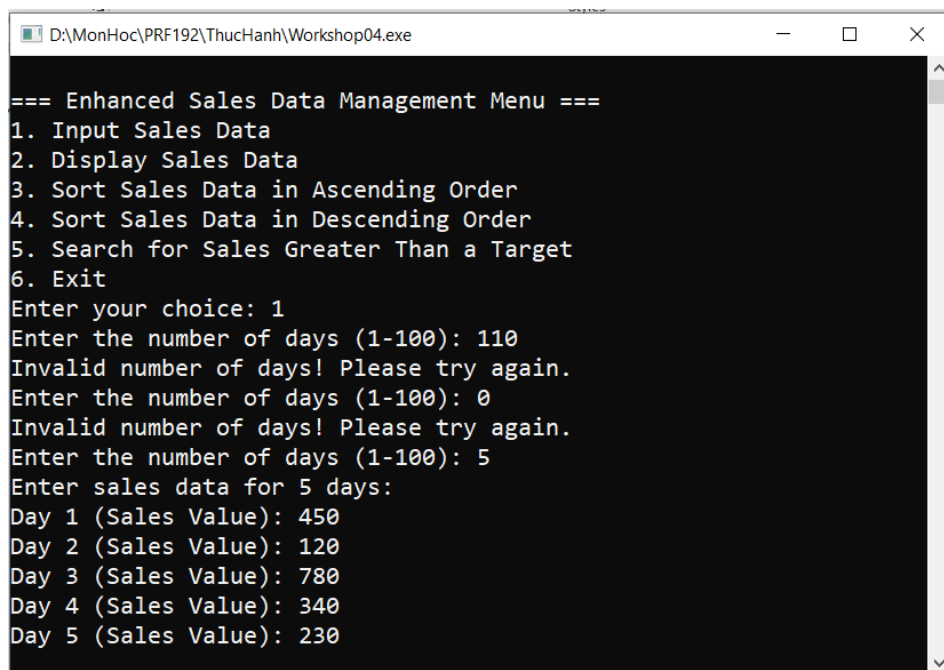
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // Constants
5  #define MAX_DAYS 100
6
7  // Struct definition
8  typedef struct {
9      int sales[MAX_DAYS];
10     int days[MAX_DAYS];
11     int count;
12 } SalesData;
13
14 // Function prototypes
15 SalesData inputSales();
16 void displaySales(SalesData data);
17 SalesData sortAscending(SalesData data);
18 SalesData sortDescending(SalesData data);
19 void searchGreaterThan(SalesData data, int target);
20
21 int main() {
22     SalesData data;
23     int choice, target;
24
25     do {
26         printf("\n=== Enhanced Sales Data Management Menu ===\n");
27         printf("1. Input Sales Data\n");
28         printf("2. Display Sales Data\n");
29         printf("3. Sort Sales Data in Ascending Order\n");
30         printf("4. Sort Sales Data in Descending Order\n");
31         printf("5. Search for Sales Greater Than a Target\n");
32         printf("6. Exit\n");
33         printf("Enter your choice: ");
34         scanf("%d", &choice);
35
36         switch (choice) {
37             case 1:
38                 data = inputSales();
39                 break;
40             case 2:
41                 displaySales(data);
42                 break;
43             case 3:
44                 data = sortAscending(data);
45                 printf("Sales data sorted in ascending order:\n");
46                 displaySales(data);
47                 break;
48             case 4:
49                 data = sortDescending(data);
50                 printf("Sales data sorted in descending order:\n");
51                 displaySales(data);
52                 break;
53             case 5:
54                 printf("Enter the target sales value: ");
55                 scanf("%d", &target);
56                 searchGreaterThan(data, target);
57                 break;
58             case 6:
59                 printf("Exiting program.\n");
60                 break;
61             default:
62                 printf("Invalid choice! Try again.\n");
63         }
64     } while (choice != 6);
65
66     system("pause");
67     return 0;
68 }

```

```
69
70 // Function to input sales data
71 SalesData inputSales() {
72     // ...
73 }
74
75 // Function to display sales data
76 void displaySales(SalesData data) {
77     // ...
78 }
79
80 // Function to sort sales data in ascending order
81 SalesData sortAscending(SalesData data) {
82     // ...
83 }
84
85 // Function to sort sales data in descending order
86 SalesData sortDescending(SalesData data) {
87     // ...
88 }
89
90 // Function to search for values greater than the target
91 void searchGreaterThan(SalesData data, int target) {
92     // ...
93 }
```

Output Sample:

1. Input Sales Data



```
D:\MonHoc\PRF192\ThucHanh\Workshop04.exe

=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 1
Enter the number of days (1-100): 110
Invalid number of days! Please try again.
Enter the number of days (1-100): 0
Invalid number of days! Please try again.
Enter the number of days (1-100): 5
Enter sales data for 5 days:
Day 1 (Sales Value): 450
Day 2 (Sales Value): 120
Day 3 (Sales Value): 780
Day 4 (Sales Value): 340
Day 5 (Sales Value): 230
```

2. Display Sales Data

```

D:\MonHoc\PRF192\ThucHanh\Workshop04.exe
=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 2

Sales data:
Day 1: 450
Day 2: 120
Day 3: 780
Day 4: 340
Day 5: 230

```

3. Sort Sales Data in Ascending Order

```

D:\MonHoc\PRF192\ThucHanh\Workshop04.exe
=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 3
Sales data sorted in ascending order:

Sales data:
Day 2: 120
Day 5: 230
Day 4: 340
Day 1: 450
Day 3: 780

```

4. Sort Sales Data in Descending Order

```

D:\MonHoc\PRF192\ThucHanh\Workshop04.exe
=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 4
Sales data sorted in descending order:

Sales data:
Day 3: 780
Day 1: 450
Day 4: 340
Day 5: 230
Day 2: 120

```

5. Search for Sales Greater Than a Target

```

D:\MonHoc\PRF192\ThucHanh\Workshop04.exe
=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 5
Enter the target sales value: 300
Sales values greater than 300:
Day 3: 780
Day 1: 450
Day 4: 340

```

6. Exit Program

```

D:\MonHoc\PRF192\ThucHanh\Workshop04.exe
=== Enhanced Sales Data Management Menu ===
1. Input Sales Data
2. Display Sales Data
3. Sort Sales Data in Ascending Order
4. Sort Sales Data in Descending Order
5. Search for Sales Greater Than a Target
6. Exit
Enter your choice: 6
Exiting program.
Press any key to continue . . .

```