

`1. Library Book Management System`

`Scenario:`

A `library system` maintains `N books` with titles. The system should:

- Sort books alphabetically
- Search for a `specific book`

Use `string handling functions` and `sorting techniques`.

`Input Example:`

...

Enter number of books: 4

Book Titles:

1. Data Structures
2. C Programming
3. Machine Learning
4. Artificial Intelligence

Sorted List:

1. Artificial Intelligence
2. C Programming
3. Data Structures
4. Machine Learning

Enter book title to search: Machine Learning

Book found in the library.

...

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

#define MAX_BOOKS 100
#define MAX_TITLE 100

int main() {
    char books[MAX_BOOKS][MAX_TITLE], temp[MAX_TITLE], search[MAX_TITLE];
    int n, i, j, found = 0;

    printf("Enter number of books: ");
    scanf("%d", &n);
    getchar();

    printf("Enter book titles:\n");
    for(i = 0; i < n; i++) {
```

```

        fgets(books[i], MAX_TITLE, stdin);
        books[i][strcspn(books[i], "\n")] = 0;
    }

    // Simple bubble sort
    for(i = 0; i < n-1; i++) {
        for(j = i+1; j < n; j++) {
            if(strcmp(books[i], books[j]) > 0) {
                strcpy(temp, books[i]);
                strcpy(books[i], books[j]);
                strcpy(books[j], temp);
            }
        }
    }

    printf("\nSorted Books:\n");
    for(i = 0; i < n; i++) {
        printf("%d. %s\n", i+1, books[i]);
    }

    printf("\nEnter book to search: ");
    fgets(search, MAX_TITLE, stdin);
    search[strcspn(search, "\n")] = 0;

    for(i = 0; i < n; i++) {
        if(strcmp(books[i], search) == 0) {
            found = 1;
            break;
        }
    }

    printf(found ? "Book found!\n" : "Book not found.\n");

```

Enter number of books: 3

Enter book titles:

AI

ML

DS

Sorted Books:

1. AI
2. DS
3. ML

Enter book to search: DS

Book found!

`2. Hospital Emergency Room System`

`Scenario:`

A `hospital ER` maintains `N` patient names
with emergency levels`

(1 to 5, where 5 is the highest). The system should:

- Sort patients based on `emergency level (descending order)`
- Display the `top 3 critical patients`

Use `arrays and sorting`.

`Input Example:`

...

Enter number of patients: 5

Patient Data:

1. Alice - Level 3
2. Bob - Level 5
3. Charlie - Level 2
4. David - Level 4
5. Eve - Level 5

Sorted by Emergency Level:

1. Bob - Level 5
2. Eve - Level 5
3. David - Level 4
4. Alice - Level 3
5. Charlie - Level 2

Top 3 Critical Patients:

1. Bob
2. Eve
3. David

...

```
include <stdio.h>

struct Patient {
    char name[50];
    int level;
```

```

};

int main() {
    int n;
    printf("Enter number of patients: ");
    scanf("%d", &n);

    struct Patient patients[n], temp;

    printf("Enter patient data:\n");
    for(int i = 0; i < n; i++) {
        printf("%d. Name: ", i+1);
        scanf("%s", patients[i].name);
        printf("    Level (1-5): ");
        scanf("%d", &patients[i].level);
    }

    for(int i = 0; i < n-1; i++) {
        for(int j = i+1; j < n; j++) {
            if(patients[i].level < patients[j].level) {
                temp = patients[i];
                patients[i] = patients[j];
                patients[j] = temp;
            }
        }
    }

    printf("\nSorted by Emergency Level:\n");
    for(int i = 0; i < n; i++) {
        printf("%d. %s - Level %d\n", i+1, patients[i].name,
patients[i].level);
    }

    printf("\nTop 3 Critical Patients:\n");
    for(int i = 0; i < 3 && i < n; i++) {
        printf("%d. %s\n", i+1, patients[i].name);
    }

    return 0;
}

```

Enter number of patients: 3

Enter patient data:

1. Name: ryan
Level (1-5): 5
2. Name: rohit
Level (1-5): 4
3. Name: rahul
Level (1-5): 3

Sorted by Emergency Level:

1. ryan - Level 5
2. rohit - Level 4
3. rahul - Level 3

Top 3 Critical Patients:

1. ryan
2. rohit
3. rahul

`3. Employee Salary Statistics`

`Scenario:`

A company records `N` employee salaries`. The system should:

- Sort salaries in `ascending order`
- Find the `second highest salary`
- Calculate the `total payroll`

Use `arrays and sorting techniques`.

`Input Example:`

...

Enter salaries of 5 employees: 35000 42000 28000 50000 39000

...

`Output Example:`

...

Sorted Salaries: 28000 35000 39000 42000 50000

Second Highest Salary: 42000

Total Payroll: ₹1,94,000

...

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

```
int main() {  
    int n;
```

```

printf("Enter number of employees: ");
scanf("%d", &n);

int salaries[n], temp, total = 0;

printf("Enter salaries: ");
for(int i = 0; i < n; i++) {
    scanf("%d", &salaries[i]);
    total += salaries[i];
}

// Sort salaries in ascending order
for(int i = 0; i < n-1; i++) {
    for(int j = i+1; j < n; j++) {
        if(salaries[i] > salaries[j]) {
            temp = salaries[i];
            salaries[i] = salaries[j];
            salaries[j] = temp;
        }
    }
}

printf("\nSorted Salaries: ");
for(int i = 0; i < n; i++) {
    printf("%d ", salaries[i]);
}

int secondHighest = salaries[n-2];
printf("\nSecond Highest Salary: %d", secondHighest);

printf("\nTotal Payroll: ₹%d\n", total);

```

Enter number of employees: 2

Enter salaries: 1000

30000

Sorted Salaries: 1000 30000

Second Highest Salary: 1000

Total Payroll: ₹31000

