```c
#include <stdio.h>

struct BankAccount {
    int accountNumber;
    char accountHolder[50];
    float balance;
};


void createAccount(struct BankAccount *account) {
    printf("Enter Account Number: ");
    scanf("%d", &account->accountNumber);
    printf("Enter Account Holder Name: ");
    scanf("%s", account->accountHolder);
    printf("Enter Initial Balance (in INR): ");
    scanf("%f", &account->balance);
    printf("Account created successfully!\n");
}


void withdraw(struct BankAccount *account) {
    float amount;
    printf("Enter Withdrawal Amount (in INR): ");
    scanf("%f", &amount);

    if (amount > 0 && amount <= account->balance) {
        account->balance -= amount;
        printf("Withdrawal successful. Updated Balance: %.2f INR\n", account
```

```c
        printf("Withdrawal successful. Updated Balance: %.2f INR\n", account
            ->balance);
    } else {
        printf("Invalid withdrawal amount or insufficient balance\n");
    }
}


void deposit(struct BankAccount *account) {
    float amount;
    printf("Enter Deposit Amount (in INR): ");
    scanf("%f", &amount);

    if (amount > 0) {
        account->balance += amount;
        printf("Deposit successful. Updated Balance: %.2f INR\n", account->balance
            );
    } else {
        printf("Invalid deposit amount\n");
    }
}


void balanceInquiry(struct BankAccount *account) {
    printf("Account Number: %d\n", account->accountNumber);
    printf("Account Holder: %s\n", account->accountHolder);
    printf("Current Balance: %.2f INR\n", account->balance);
}
```

```
54
55  int main() {
56      struct BankAccount userAccount;
57      int choice;
58
59      do {
60          printf("\n*** Basic Banking System Menu ***\n");
61          printf("1. Create Account\n");
62          printf("2. Withdraw\n");
63          printf("3. Deposit\n");
64          printf("4. Balance Inquiry\n");
65          printf("0. Exit\n");
66          printf("Enter your choice: ");
67          scanf("%d", &choice);
68
69          switch (choice) {
70              case 1:
71                  createAccount(&userAccount);
72                  break;
73              case 2:
74                  withdraw(&userAccount);
75                  break;
76              case 3:
77                  deposit(&userAccount);
78                  break;
79              case 4:
80                  balanceInquiry(&userAccount);
81                  break;
```

```
81                  break;
82              case 0:
83                  printf("Exiting the program. Goodbye!\n");
84                  break;
85              default:
86                  printf("Invalid choice. Please enter a valid option.\n");
87          }
88      } while (choice != 0);
89
90      return 0;
91  }
92
```

```
*** Basic Banking System Menu ***
1. Create Account
2. Withdraw
3. Deposit
4. Balance Inquiry
0. Exit
Enter your choice: 1
Enter Account Number: 123456789
Enter Account Holder Name: pooja gaikwad
Enter Initial Balance (in INR): Account created successfully!

*** Basic Banking System Menu ***
1. Create Account
2. Withdraw
3. Deposit
4. Balance Inquiry
0. Exit
Enter your choice: Enter Account Number: Enter Account Holder Name: Enter Initial
    Balance (in INR): 4
```

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

#define MAX_STRINGS 10
#define MAX_LENGTH 100

void swapStrings(char *a, char *b) {
    char temp[MAX_LENGTH];
    strcpy(temp, a);
    strcpy(a, b);
    strcpy(b, temp);
}

void sortStrings(char strings[MAX_STRINGS][MAX_LENGTH], int numStrings) {
    for (int i = 0; i < numStrings - 1; i++) {
        for (int j = i + 1; j < numStrings; j++) {
            if (strcmp(strings[i], strings[j]) > 0) {
                swapStrings(strings[i], strings[j]);
            }
        }
    }
}

int main() {
    char strings[MAX_STRINGS][MAX_LENGTH];
    int numStrings;

    printf("Enter the number of strings (up to %d): ", MAX_STRINGS);
```

```c
    scanf("%d", &numStrings);

    printf("Enter %d strings:\n", numStrings);
    for (int i = 0; i < numStrings; i++) {
        scanf("%s", strings[i]);
    }

    sortStrings(strings, numStrings);

    printf("\nSorted strings:\n");
    for (int i = 0; i < numStrings; i++) {
        printf("%s\n", strings[i]);
    }

    return 0;
}
```

```
Enter the number of strings (up to 10): 4
Enter 4 strings:
pooja
tannu
nidhi
ritika
Sorted strings:
nidhi
pooja
ritika
tannu
```

```c
#include <stdio.h>


int isElementPresent(int arr[10][10], int rows, int cols, int target) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (arr[i][j] == target) {
                return 1;
            }
        }
    }
    return 0;
}

int main() {
    int rows, cols, target;


    printf("Enter the number of rows and columns (up to 10 each): ");
    scanf("%d %d", &rows, &cols);


    int arr[10][10];
    printf("Enter the elements of the 2D array:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Element at position [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
```

```c
    scanf("%d %d", &rows, &cols);

    int arr[10][10];
    printf("Enter the elements of the 2D array:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Element at position [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    printf("Enter the element to search: ");
    scanf("%d", &target);

    if (isElementPresent(arr, rows, cols, target)) {
        printf("Element %d is present in the 2D array.\n", target);
    } else {
        printf("Element %d is not present in the 2D array.\n", target);
    }

    return 0;
}
```

```
Enter the number of rows and columns (up to 10 each): 2 3
Enter the elements of the 2D array:
Element at position [0][0]: 1
Element at position [0][1]: 2
Element at position [0][2]: 3
Element at position [1][0]: 4
Element at position [1][1]: 5
Element at position [1][2]: 6
Enter the element to search: 5
Element 5 is present in the 2D array.
```

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

int searchSubstring(const char *haystack, const char *needle) {
    int haystackLen = strlen(haystack);
    int needleLen = strlen(needle);

    for (int i = 0; i <= haystackLen - needleLen; i++) {
        int j;
        for (j = 0; j < needleLen; j++) {
            if (haystack[i + j] != needle[j]) {
                break;
            }
        }


        if (j == needleLen) {
            return i;
        }
    }

    return -1;
}

int main() {
    char largerString[100];
    char substring[50];
```

```c
int main() {
    char largerString[100];
    char substring[50];


    printf("Enter the larger string: ");
    fgets(largerString, sizeof(largerString), stdin);
    largerString[strcspn(largerString, "\n")] = '\0';


    printf("Enter the substring to search for: ");
    fgets(substring, sizeof(substring), stdin);
    substring[strcspn(substring, "\n")] = '\0';


    int index = searchSubstring(largerString, substring);


    if (index != -1) {
        printf("Substring found at index %d\n", index);
    } else {
        printf("Substring not found\n");
    }

    return 0;
```

```
Enter the larger string: pooja gaikwad
Enter the substring to search for: gai
Substring found at index 6
```

```c
#include <stdio.h>


int lastIndexOf(int array[], int size, int target) {
    int lastIndex = -1;

    for (int i = 0; i < size; i++) {
        if (array[i] == target) {
            lastIndex = i;
        }
    }

    return lastIndex;
}

int main() {
    int size, target;


    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int array[size]

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &array[i]);
```

```
    scanf("%d", &array[i]);
}

printf("Enter the number to search for: ");
scanf("%d", &target);


int lastIndex = lastIndexOf(array, size, target);

if (lastIndex != -1) {
    printf("Last occurrence of %d is at index %d\n", target, lastIndex);
} else {
    printf("%d not found in the array\n", target);
}

return 0;
```

```
Enter the size of the array: 5
Enter the array elements:
1 2 3 2 4
Enter the number to find: 2
Last occurrence of 2 is at index 3
```

```c
#include <stdio.h>
int linearSearch(int arr[], int size, int key) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}

int main() {
    int size, key;

    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];

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

    printf("Enter the element to search: ");
    scanf("%d", &key);


    int index = linearSearch(arr, size, key);
```

```c
int main() {
    int size, key;

    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];

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

    printf("Enter the element to search: ");
    scanf("%d", &key);


    int index = linearSearch(arr, size, key);

    if (index != -1) {
        printf("Element %d found at index %d.\n", key, index);
    } else {
        printf("Element %d not found in the array.\n", key);
    }

    return 0;
}
```

```
Enter the size of the array: 5
Enter 5 elements:
10 20 30 40 50
Enter the element to search: 30
Element 30 found at index 2.
```

```c
#include <stdio.h>

int binarySearch(int arr[], int size, int target) {
    int left = 0, right = size - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;


        if (arr[mid] == target)
            return mid;

        if (arr[mid] < target)
            left = mid + 1;


        else
            right = mid - 1;
    }


    return -1;
}

int main() {
    int size, target;
```

```c
printf("Enter the size of the array: ");
scanf("%d", &size);

int arr[size];


printf("Enter the elements of the array in sorted order:\n");
for (int i = 0; i < size; i++) {
    scanf("%d", &arr[i]);
}


printf("Enter the element to be searched: ");
scanf("%d", &target);


int result = binarySearch(arr, size, target);


if (result != -1) {
    printf("Element %d found at index %d\n", target, result);
} else {
    printf("Element %d not found in the array\n", target);
}

return 0;
```

```
Enter the size of the array: 5
Enter the elements of the array in sorted order:
1 2 3 4 5
Enter the element to be searched: 3
Element 3 found at index 2
```

```c
#include <stdio.h>

void findMinMax(int arr[], int size, int *min, int *max) {

    *min = *max = arr[0];


    for (int i = 1; i < size; ++i) {
        if (arr[i] < *min) {
            *min = arr[i];
        } else if (arr[i] > *max) {
            *max = arr[i];
        }
    }
}

int main() {
    int size;


    printf("Enter the size of the array: ");
    scanf("%d", &size);


    int arr[size];
```

```c
printf("Enter the size of the array: ");
scanf("%d", &size);


int arr[size];


printf("Enter %d elements:\n", size);
for (int i = 0; i < size; ++i) {
    printf("Element %d: ", i + 1);
    scanf("%d", &arr[i]);
}


int min, max;


findMinMax(arr, size, &min, &max);


printf("\nMinimum element: %d\n", min);
printf("Maximum element: %d\n", max);

return 0;
```

```
Enter the size of the array: 5
Enter 5 elements:
Element 1: 1 2 3 4 5
Element 2: Element 3: Element 4: Element 5:
Minimum element: 1
Maximum element: 5
```