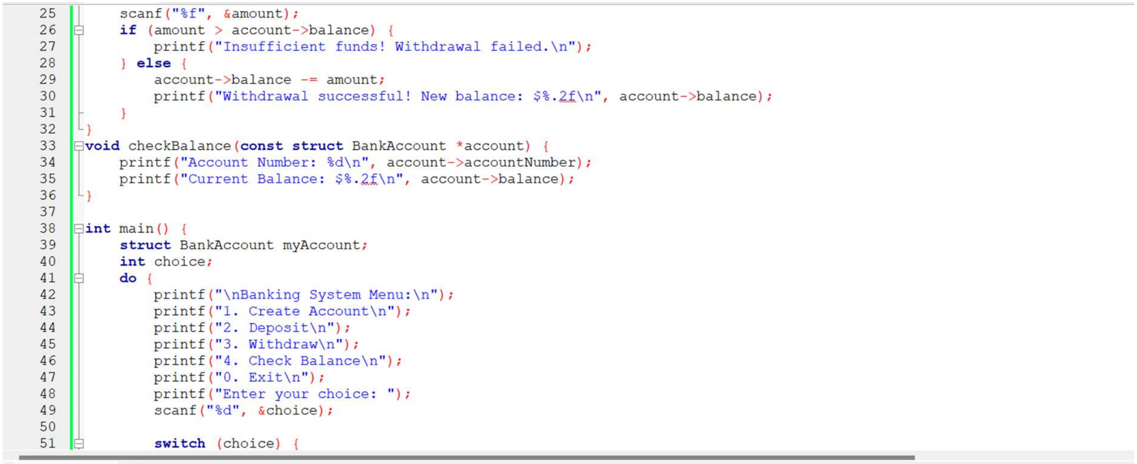
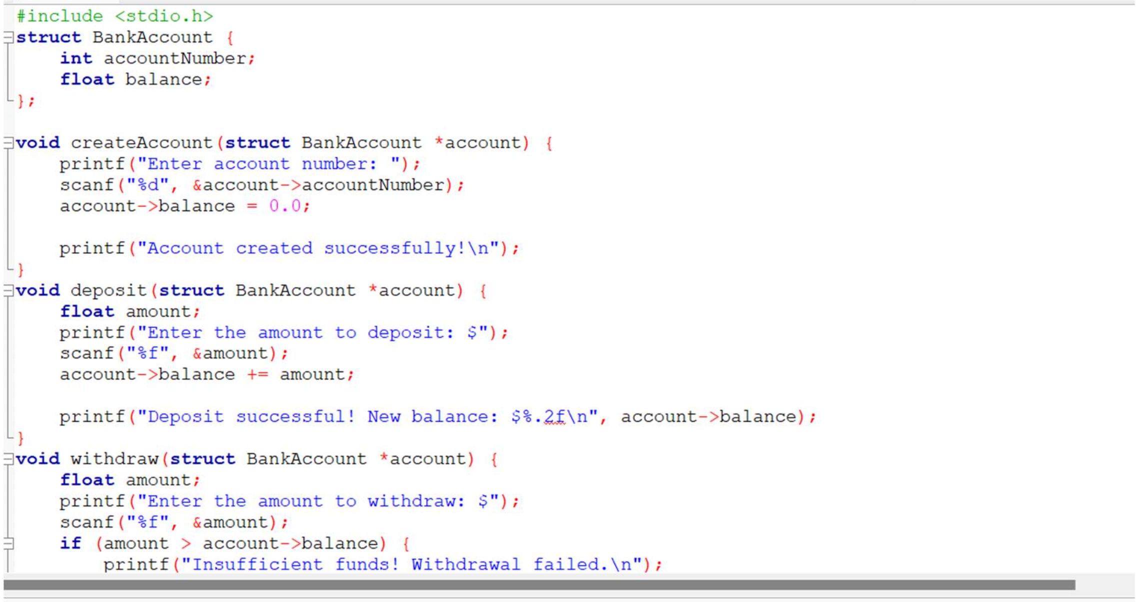
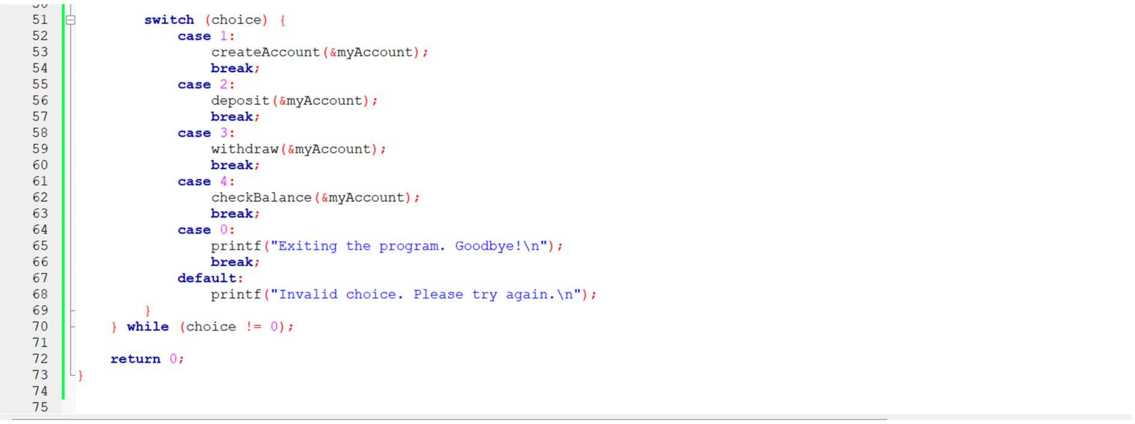
Program-1



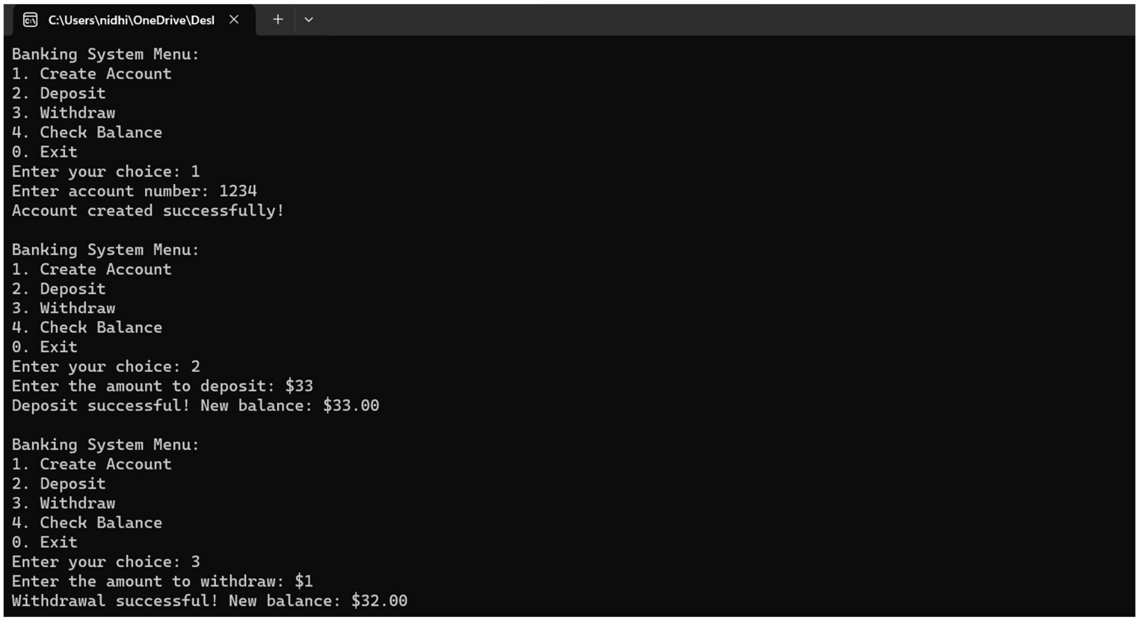
```
    scanf("%f", &amount);
    if (amount > account->balance) {
        printf("Insufficient funds! Withdrawal failed.\n");
    } else {
        account->balance -= amount;
        printf("Withdrawal successful! New balance: $%.2f\n", account->balance);
    }
}
void checkBalance(const struct BankAccount *account) {
    printf("Account Number: %d\n", account->accountNumber);
    printf("Current Balance: $%.2f\n", account->balance);
}

int main() {
    struct BankAccount myAccount;
    int choice;
    do {
        printf("\nBanking System Menu:\n");
        printf("1. Create Account\n");
        printf("2. Deposit\n");
        printf("3. Withdraw\n");
        printf("4. Check Balance\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
```



```
#include <stdio.h>
struct BankAccount {
    int accountNumber;
    float balance;
};

void createAccount(struct BankAccount *account) {
    printf("Enter account number: ");
    scanf("%d", &account->accountNumber);
    account->balance = 0.0;

    printf("Account created successfully!\n");
}
void deposit(struct BankAccount *account) {
    float amount;
    printf("Enter the amount to deposit: $");
    scanf("%f", &amount);
    account->balance += amount;

    printf("Deposit successful! New balance: $%.2f\n", account->balance);
}
void withdraw(struct BankAccount *account) {
    float amount;
    printf("Enter the amount to withdraw: $");
    scanf("%f", &amount);
    if (amount > account->balance) {
        printf("Insufficient funds! Withdrawal failed.\n");
```

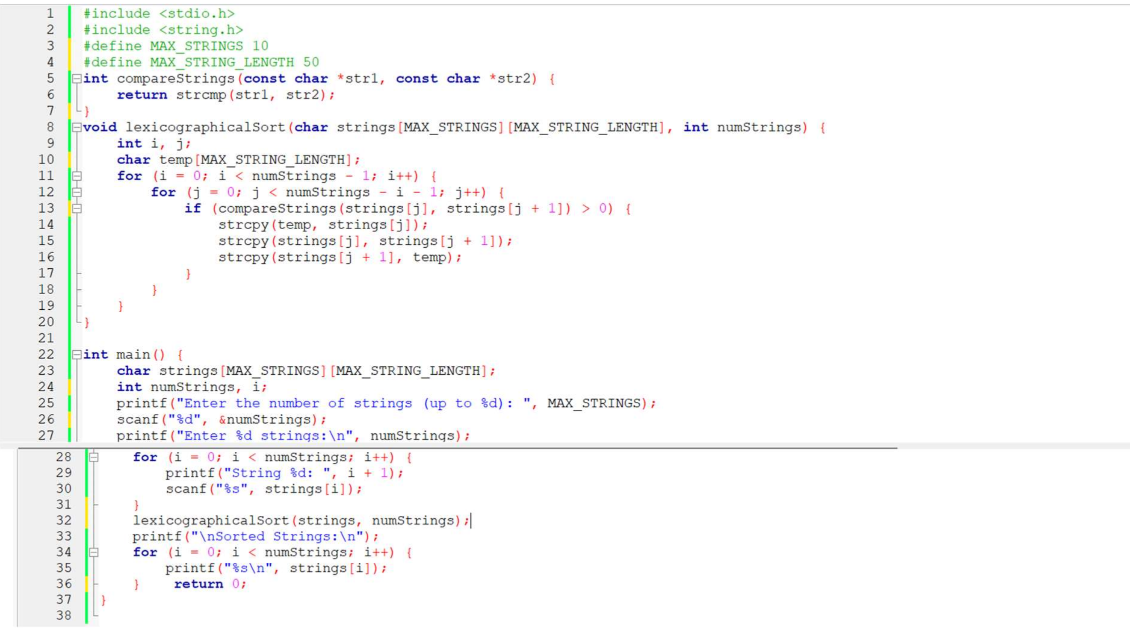


```
        switch (choice) {
            case 1:
                createAccount(&myAccount);
                break;
            case 2:
                deposit(&myAccount);
                break;
            case 3:
                withdraw(&myAccount);
                break;
            case 4:
                checkBalance(&myAccount);
                break;
            case 0:
                printf("Exiting the program. Goodbye!\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 0);

    return 0;
}
```

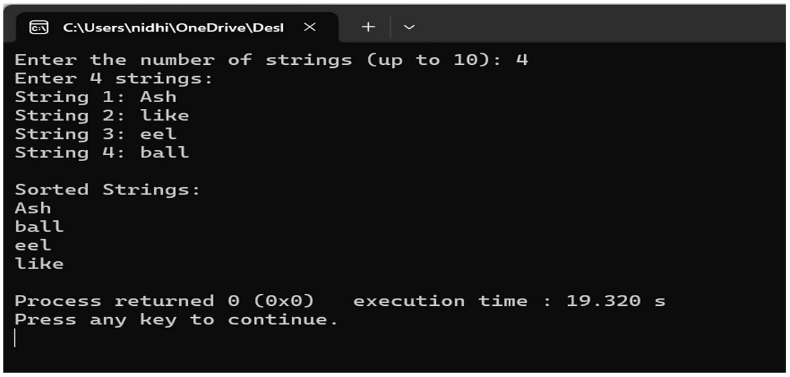


```
Banking System Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
0. Exit
Enter your choice: 1
Enter account number: 1234
Account created successfully!

Banking System Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
0. Exit
Enter your choice: 2
Enter the amount to deposit: $33
Deposit successful! New balance: $33.00

Banking System Menu:
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
0. Exit
Enter your choice: 3
Enter the amount to withdraw: $1
Withdrawal successful! New balance: $32.00
```

Program-2

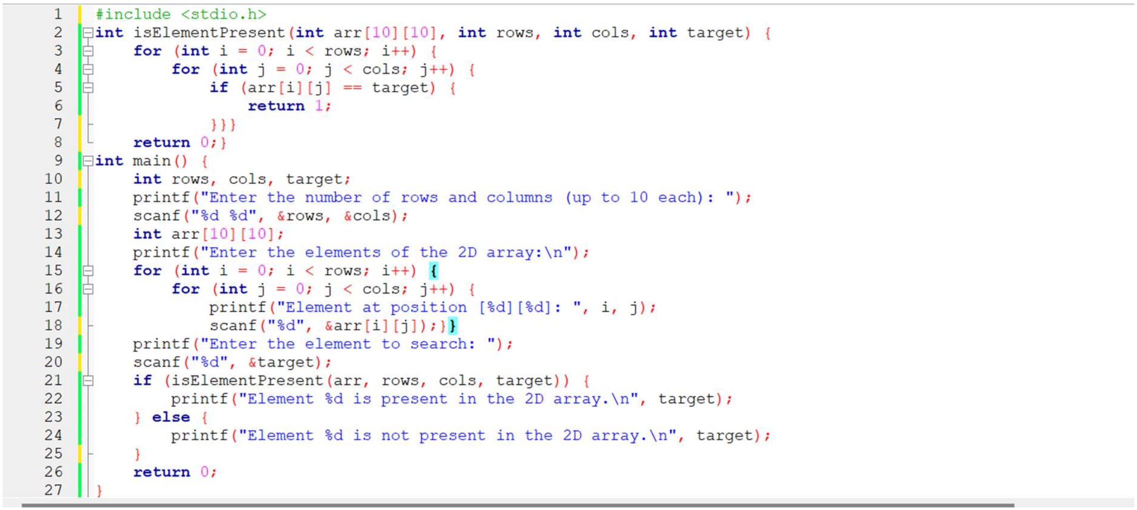


```c
#include <stdio.h>
#include <string.h>
#define MAX_STRINGS 10
#define MAX_STRING_LENGTH 50
int compareStrings(const char *str1, const char *str2) {
    return strcmp(str1, str2);
}
void lexicographicalSort(char strings[MAX_STRINGS][MAX_STRING_LENGTH], int numStrings) {
    int i, j;
    char temp[MAX_STRING_LENGTH];
    for (i = 0; i < numStrings - 1; i++) {
        for (j = 0; j < numStrings - i - 1; j++) {
            if (compareStrings(strings[j], strings[j + 1]) > 0) {
                strcpy(temp, strings[j]);
                strcpy(strings[j], strings[j + 1]);
                strcpy(strings[j + 1], temp);
            }
        }
    }
}

int main() {
    char strings[MAX_STRINGS][MAX_STRING_LENGTH];
    int numStrings, i;
    printf("Enter the number of strings (up to %d): ", MAX_STRINGS);
    scanf("%d", &numStrings);
    printf("Enter %d strings:\n", numStrings);
    for (i = 0; i < numStrings; i++) {
        printf("String %d: ", i + 1);
        scanf("%s", strings[i]);
    }
    lexicographicalSort(strings, numStrings);
    printf("\nSorted Strings:\n");
    for (i = 0; i < numStrings; i++) {
        printf("%s\n", strings[i]);
    }     return 0;
}
```

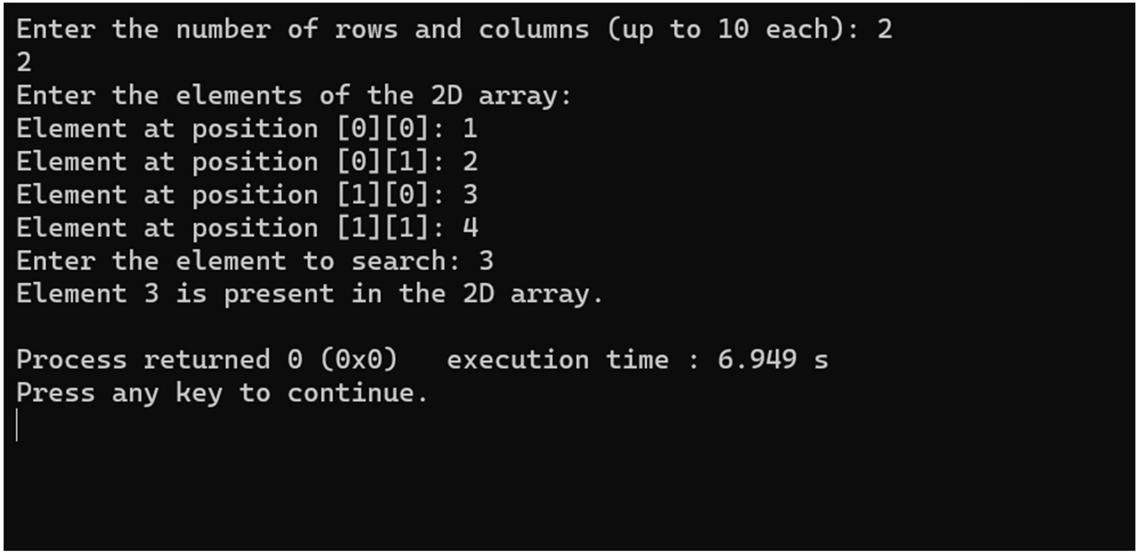


```
Enter the number of strings (up to 10): 4
Enter 4 strings:
String 1: Ash
String 2: like
String 3: eel
String 4: ball

Sorted Strings:
Ash
ball
eel
like

Process returned 0 (0x0)   execution time : 19.320 s
Press any key to continue.
```

Program-3

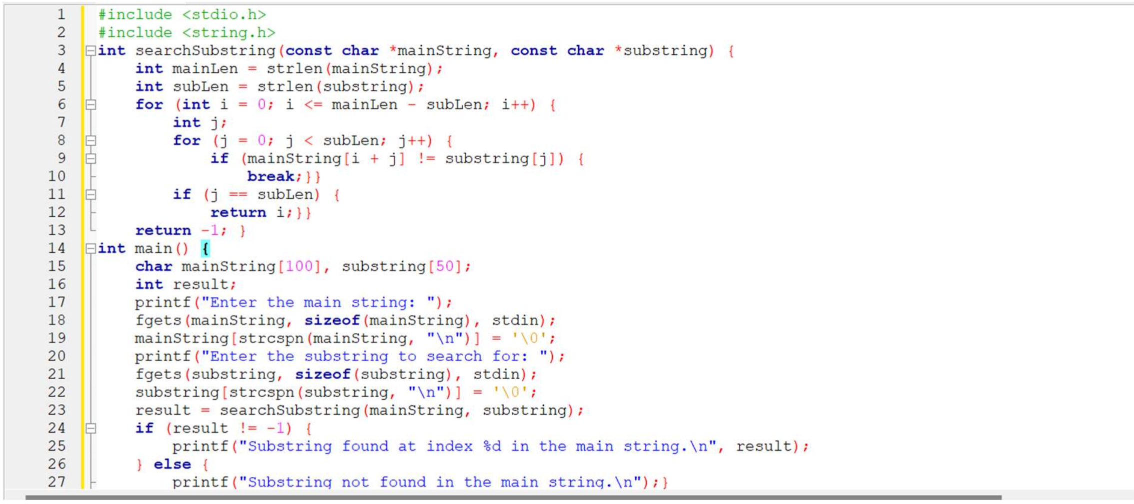


```c
#include <stdio.h>
int isElementPresent(int arr[10][10], int rows, int cols, int target) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (arr[i][j] == target) {
                return 1;
            }}}
    return 0;}
int main() {
    int rows, cols, 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]);}}
    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;
}
```

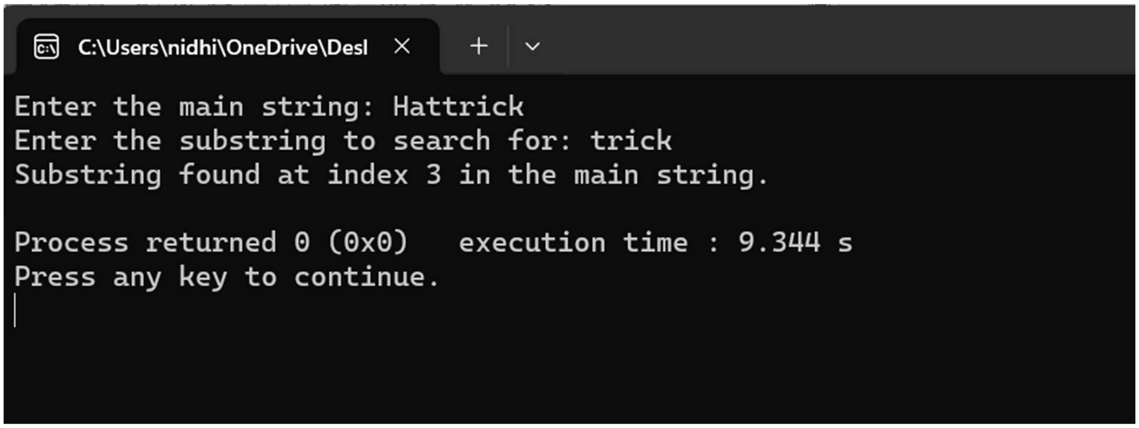


```
Enter the number of rows and columns (up to 10 each): 2
2
Enter the elements of the 2D array:
Element at position [0][0]: 1
Element at position [0][1]: 2
Element at position [1][0]: 3
Element at position [1][1]: 4
Enter the element to search: 3
Element 3 is present in the 2D array.

Process returned 0 (0x0)   execution time : 6.949 s
Press any key to continue.
```

Program-4

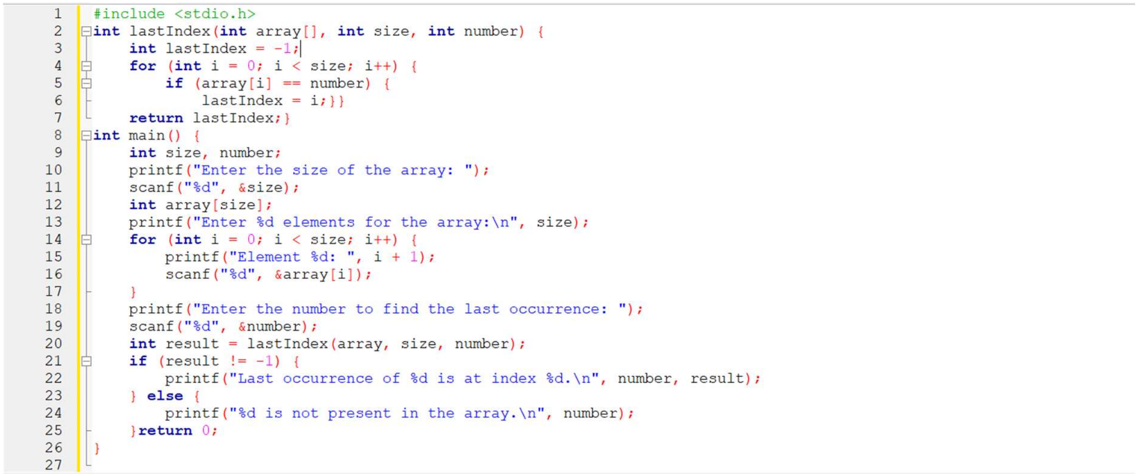


```c
#include <stdio.h>
#include <string.h>
int searchSubstring(const char *mainString, const char *substring) {
    int mainLen = strlen(mainString);
    int subLen = strlen(substring);
    for (int i = 0; i <= mainLen - subLen; i++) {
        int j;
        for (j = 0; j < subLen; j++) {
            if (mainString[i + j] != substring[j]) {
                break;}}
        if (j == subLen) {
            return i;}}
    return -1; }
int main() {
    char mainString[100], substring[50];
    int result;
    printf("Enter the main string: ");
    fgets(mainString, sizeof(mainString), stdin);
    mainString[strcspn(mainString, "\n")] = '\0';
    printf("Enter the substring to search for: ");
    fgets(substring, sizeof(substring), stdin);
    substring[strcspn(substring, "\n")] = '\0';
    result = searchSubstring(mainString, substring);
    if (result != -1) {
        printf("Substring found at index %d in the main string.\n", result);
    } else {
        printf("Substring not found in the main string.\n");}
```

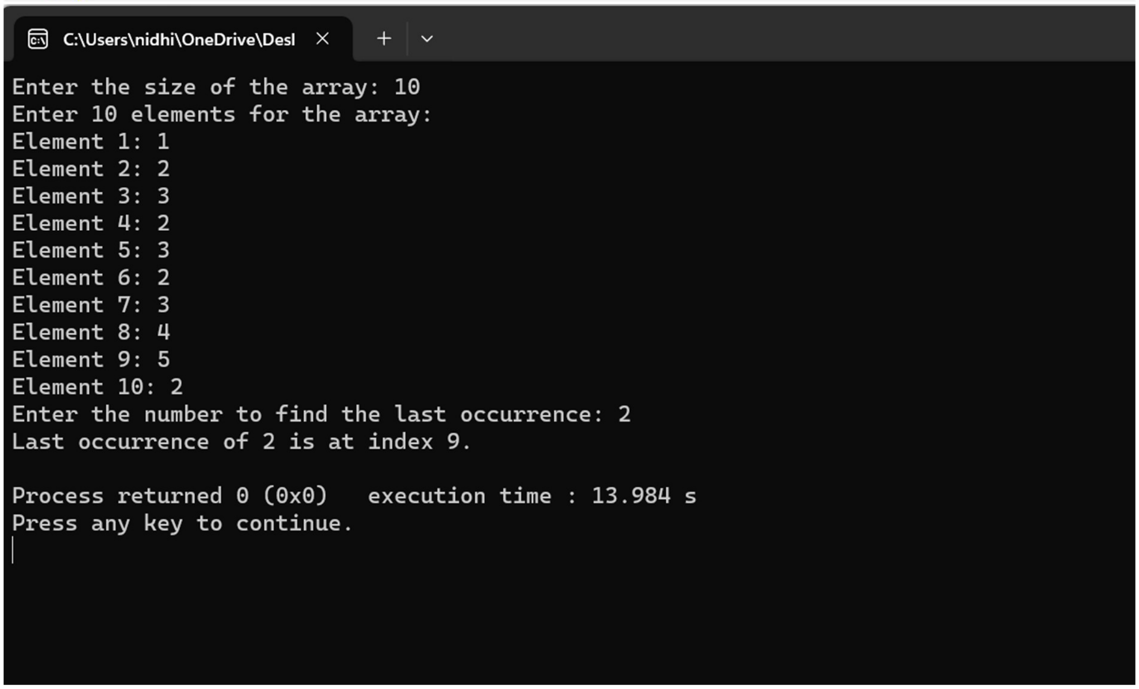


```
Enter the main string: Hattrick
Enter the substring to search for: trick
Substring found at index 3 in the main string.

Process returned 0 (0x0)   execution time : 9.344 s
Press any key to continue.
```

Program-5

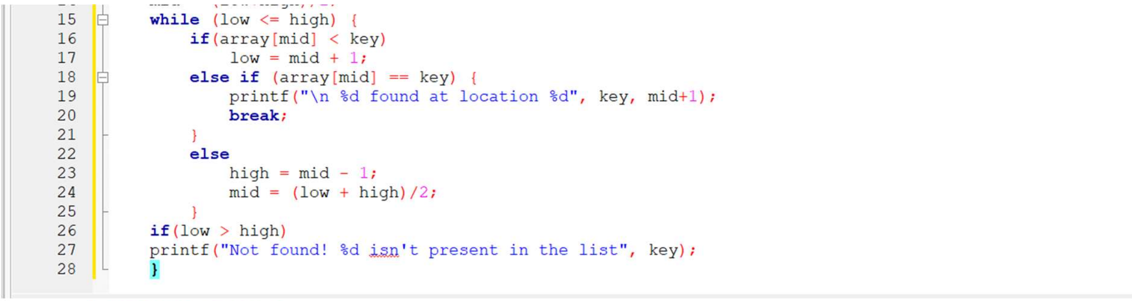


```c
#include <stdio.h>
int lastIndex(int array[], int size, int number) {
    int lastIndex = -1;
    for (int i = 0; i < size; i++) {
        if (array[i] == number) {
            lastIndex = i;}}
    return lastIndex;}
int main() {
    int size, number;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int array[size];
    printf("Enter %d elements for the array:\n", size);
    for (int i = 0; i < size; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &array[i]);
    }
    printf("Enter the number to find the last occurrence: ");
    scanf("%d", &number);
    int result = lastIndex(array, size, number);
    if (result != -1) {
        printf("Last occurrence of %d is at index %d.\n", number, result);
    } else {
        printf("%d is not present in the array.\n", number);
    }return 0;
}
```

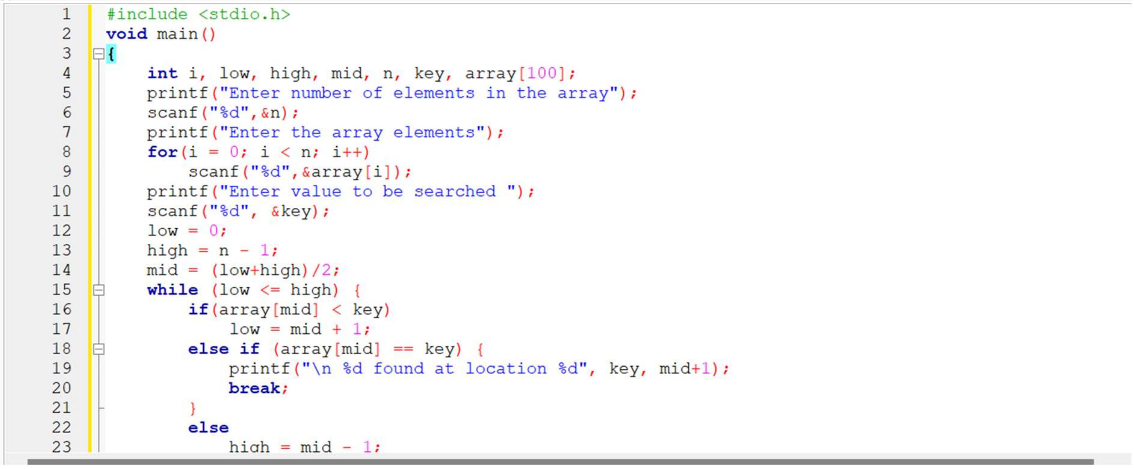


```
Enter the size of the array: 10
Enter 10 elements for the array:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 2
Element 5: 3
Element 6: 2
Element 7: 3
Element 8: 4
Element 9: 5
Element 10: 2
Enter the number to find the last occurrence: 2
Last occurrence of 2 is at index 9.

Process returned 0 (0x0)   execution time : 13.984 s
Press any key to continue.
```

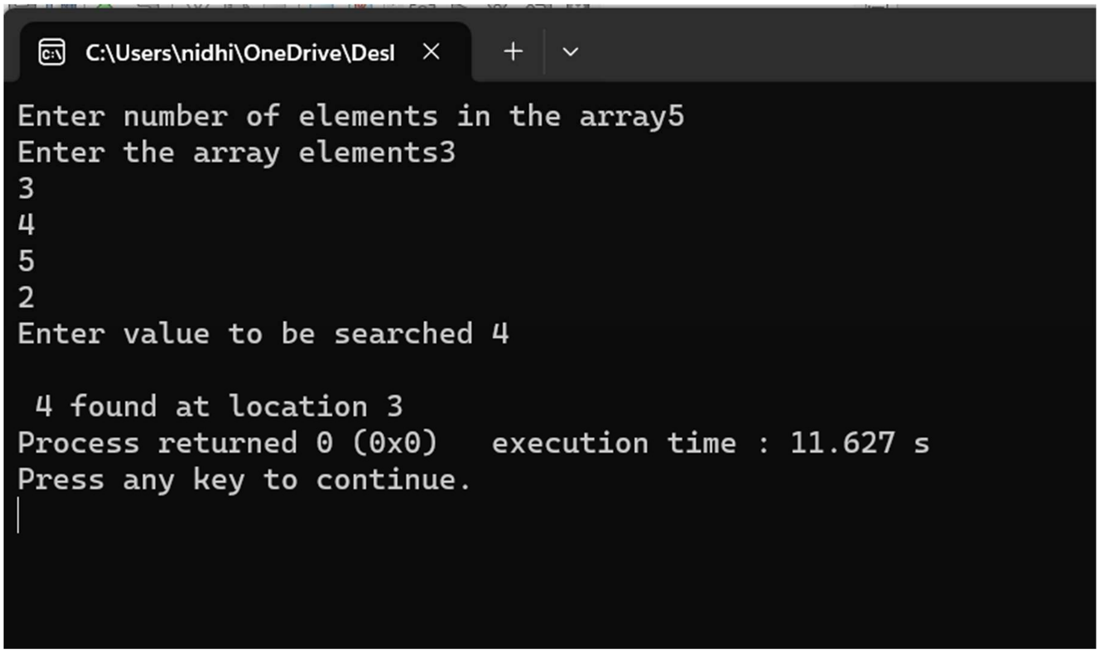
Program-6



```
    while (low <= high) {
        if(array[mid] < key)
            low = mid + 1;
        else if (array[mid] == key) {
            printf("\n %d found at location %d", key, mid+1);
            break;
        }
        else
            high = mid - 1;
            mid = (low + high)/2;
        }
    if(low > high)
    printf("Not found! %d isn't present in the list", key);
    }
```

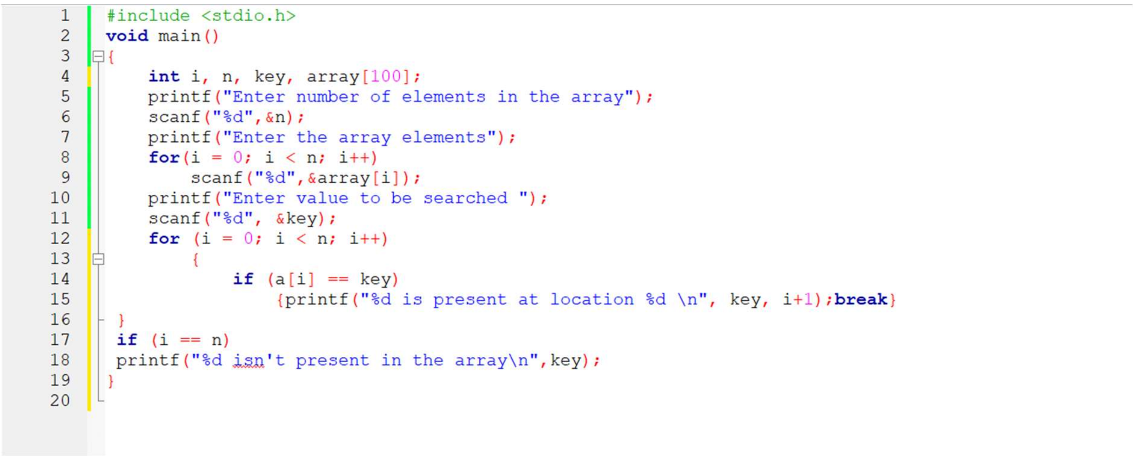


```
#include <stdio.h>
void main()
{
    int i, low, high, mid, n, key, array[100];
    printf("Enter number of elements in the array");
    scanf("%d",&n);
    printf("Enter the array elements");
    for(i = 0; i < n; i++)
        scanf("%d",&array[i]);
    printf("Enter value to be searched ");
    scanf("%d", &key);
    low = 0;
    high = n - 1;
    mid = (low+high)/2;
    while (low <= high) {
        if(array[mid] < key)
            low = mid + 1;
        else if (array[mid] == key) {
            printf("\n %d found at location %d", key, mid+1);
            break;
        }
        else
            high = mid - 1;
```

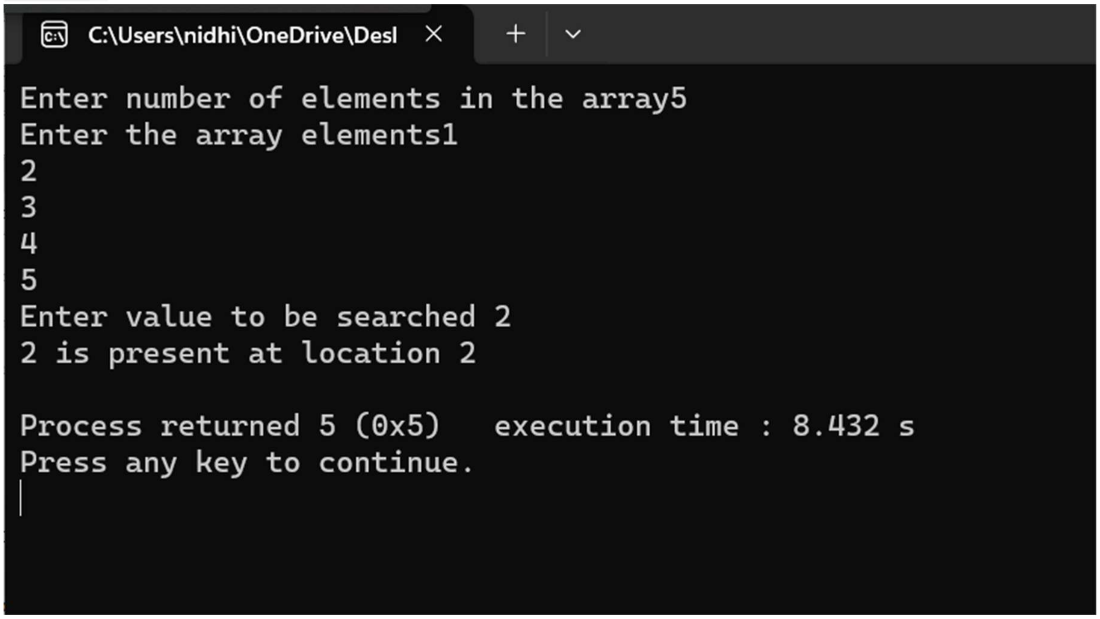


```
Enter number of elements in the array5
Enter the array elements3
3
4
5
2
Enter value to be searched 4

 4 found at location 3
Process returned 0 (0x0)   execution time : 11.627 s
Press any key to continue.
```

Program-7

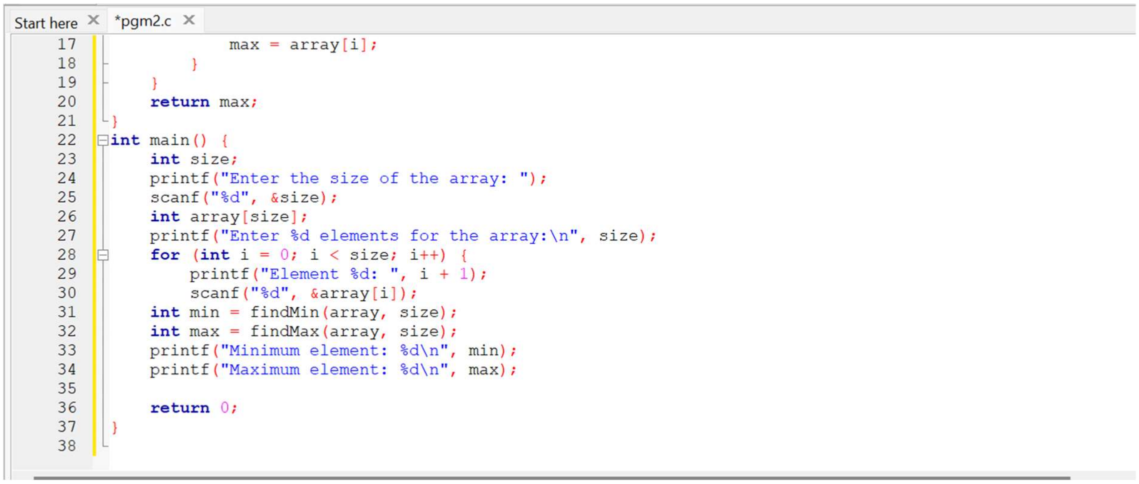


```
#include <stdio.h>
void main()
{
    int i, n, key, array[100];
    printf("Enter number of elements in the array");
    scanf("%d",&n);
    printf("Enter the array elements");
    for(i = 0; i < n; i++)
        scanf("%d",&array[i]);
    printf("Enter value to be searched ");
    scanf("%d", &key);
    for (i = 0; i < n; i++)
        {
            if (a[i] == key)
                {printf("%d is present at location %d \n", key, i+1);break}
 }
 if (i == n)
 printf("%d isn't present in the array\n",key);
}
```

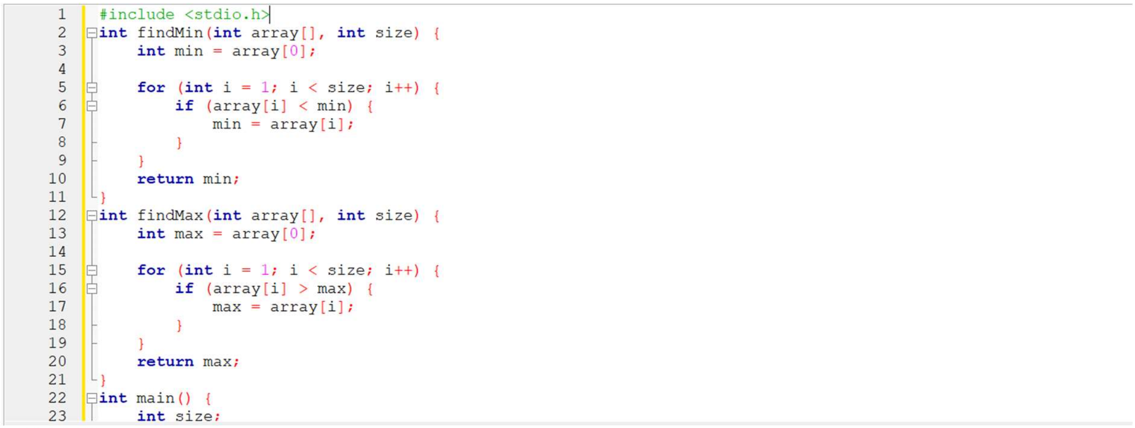


```
Enter number of elements in the array5
Enter the array elements1
2
3
4
5
Enter value to be searched 2
2 is present at location 2

Process returned 5 (0x5)   execution time : 8.432 s
Press any key to continue.
```

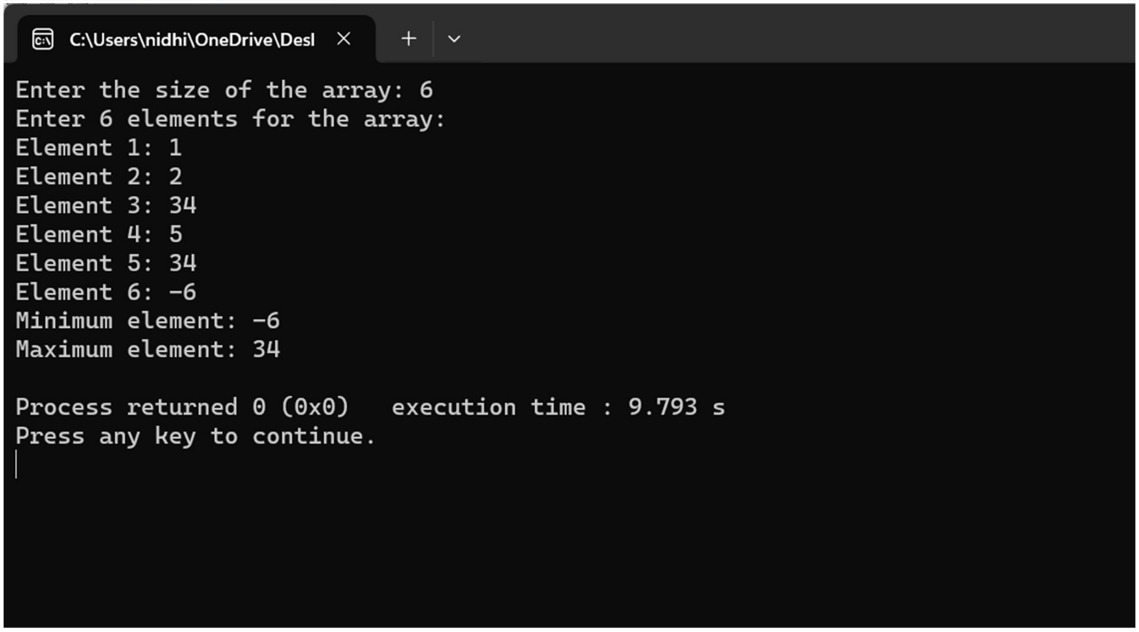
Program-8



```
            max = array[i];
        }
    }
    return max;
}
int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int array[size];
    printf("Enter %d elements for the array:\n", size);
    for (int i = 0; i < size; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &array[i]);
    int min = findMin(array, size);
    int max = findMax(array, size);
    printf("Minimum element: %d\n", min);
    printf("Maximum element: %d\n", max);

    return 0;
}
```



```c
#include <stdio.h>
int findMin(int array[], int size) {
    int min = array[0];

    for (int i = 1; i < size; i++) {
        if (array[i] < min) {
            min = array[i];
        }
    }
    return min;
}
int findMax(int array[], int size) {
    int max = array[0];

    for (int i = 1; i < size; i++) {
        if (array[i] > max) {
            max = array[i];
        }
    }
    return max;
}
int main() {
    int size;
```



```
Enter the size of the array: 6
Enter 6 elements for the array:
Element 1: 1
Element 2: 2
Element 3: 34
Element 4: 5
Element 5: 34
Element 6: -6
Minimum element: -6
Maximum element: 34

Process returned 0 (0x0)   execution time : 9.793 s
Press any key to continue.
```