1.Write a C program to perform Matrix Multiplication.

Aim:

To write a C program that performs Matrix Multiplication on two given matrices and displays the resulting matrix.

Algorithm:

```
1. Start
2. Read matrix A of size m × n
3. Read matrix B of size n × p
4. Initialize matrix C of size m × p with all zeros
5. For each row i from 0 to m-1
    For each column j from 0 to p-1
        Set C[i][j] = 0
        For each k from 0 to n-1
            C[i][j] += A[i][k] * B[k][j]
6. Display matrix C
7. End
```

Input:

Enter rows and columns of matrix A =2,3

Enter rows and columns of second matrix=3,2

Output:

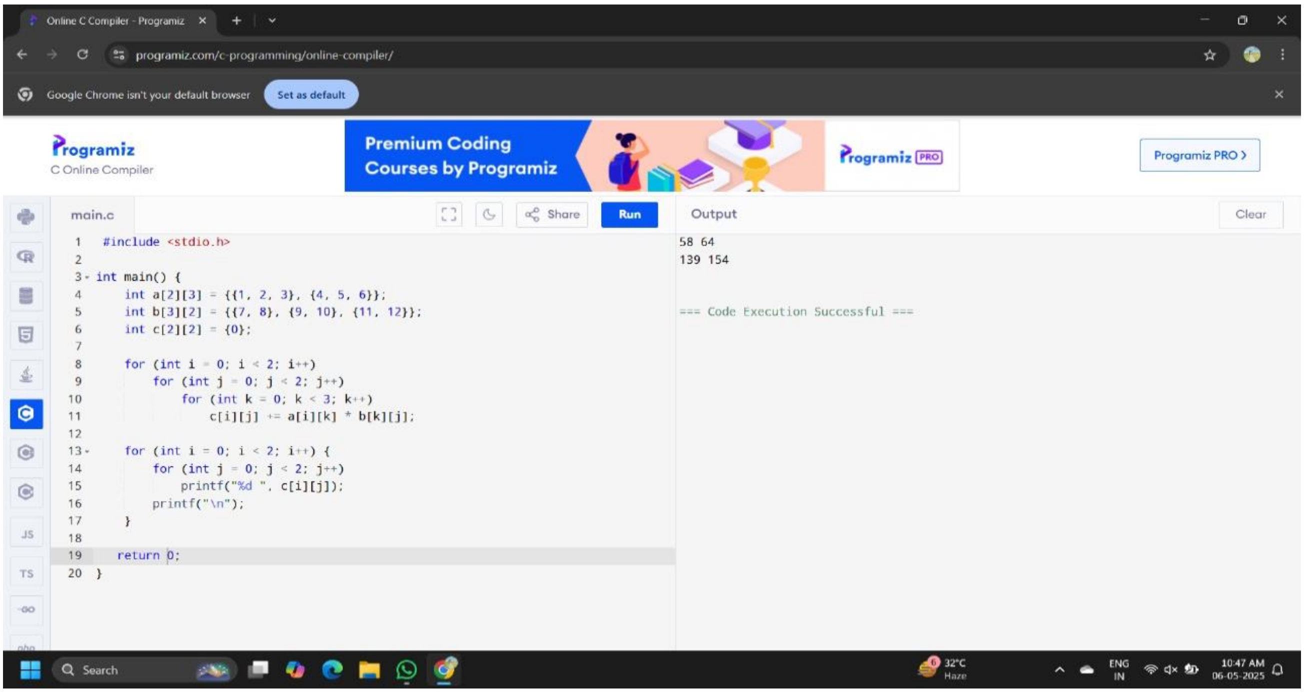
```
Enter no of elements of matrix A
1 2 3
4 5 6
Enter no of elemenets of matrix B
7 8
9 10
11 12
```

Resultant matrix after multiplication

58 64

139 154

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO >
main.c
Share
Run
Output
Clear
Search
32°C Haze
ENG IN
10:47 AM 06-05-2025

```c
#include <stdio.h>

int main() {
    int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
    int b[3][2] = {{7, 8}, {9, 10}, {11, 12}};
    int c[2][2] = {0};

    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 2; j++)
            for (int k = 0; k < 3; k++)
                c[i][j] += a[i][k] * b[k][j];

    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++)
            printf("%d ", c[i][j]);
        printf("\n");
    }

  return 0;
}
```

```
58 64
139 154


=== Code Execution Successful ===
```

2. Write a C program to find Odd or Even number from a given set of numbers.

Aim:

To write a C program that determines whether a given set of numbers are odd or even

Algorithm:

```
1. Start
2. Read the total number of elements, say n
3. For i = 0 to n - 1
    a. Read number x
    b. If x % 2 == 0
        Display "x is Even"
    c. Else
        Display "x is Odd"
```

4. End

Input:

Enter how many numbers = 5

Enter 5 numbers:

12 7 9 4 10

Output:

12 is Even

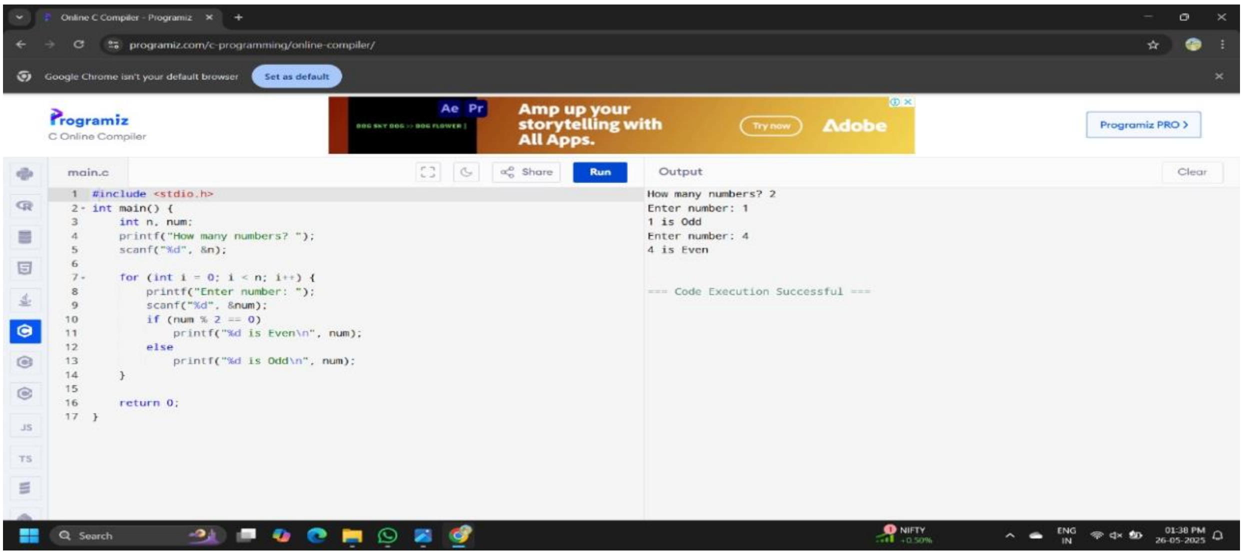
7 is odd

9 is odd

4 is Even

10 is Even

Code:


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Programiz PRO ›
main.c
Share
Run
Output
Clear
Search

```
#include <stdio.h>
int main() {
    int n, num;
    printf("How many numbers? ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        printf("Enter number: ");
        scanf("%d", &num);
        if (num % 2 == 0)
            printf("%d is Even\n", num);
        else
            printf("%d is Odd\n", num);
    }

    return 0;
}
```

```
How many numbers? 2
Enter number: 1
1 is Odd
Enter number: 4
4 is Even


=== Code Execution Successful ===
```

3. Write a C program to find Factorial of a given number without using Recursion.

Aim:

To write a C program to find the factorial of a given number using an iterative (non-recursive) approach.

Algorithm:

```
1. Start
2. Read the integer n
3. If n < 0, display "Invalid input" and stop
4. Initialize a variable fact = 1
5. For i = 1 to n
       fact = fact * i
6. Display fact
7. End
```

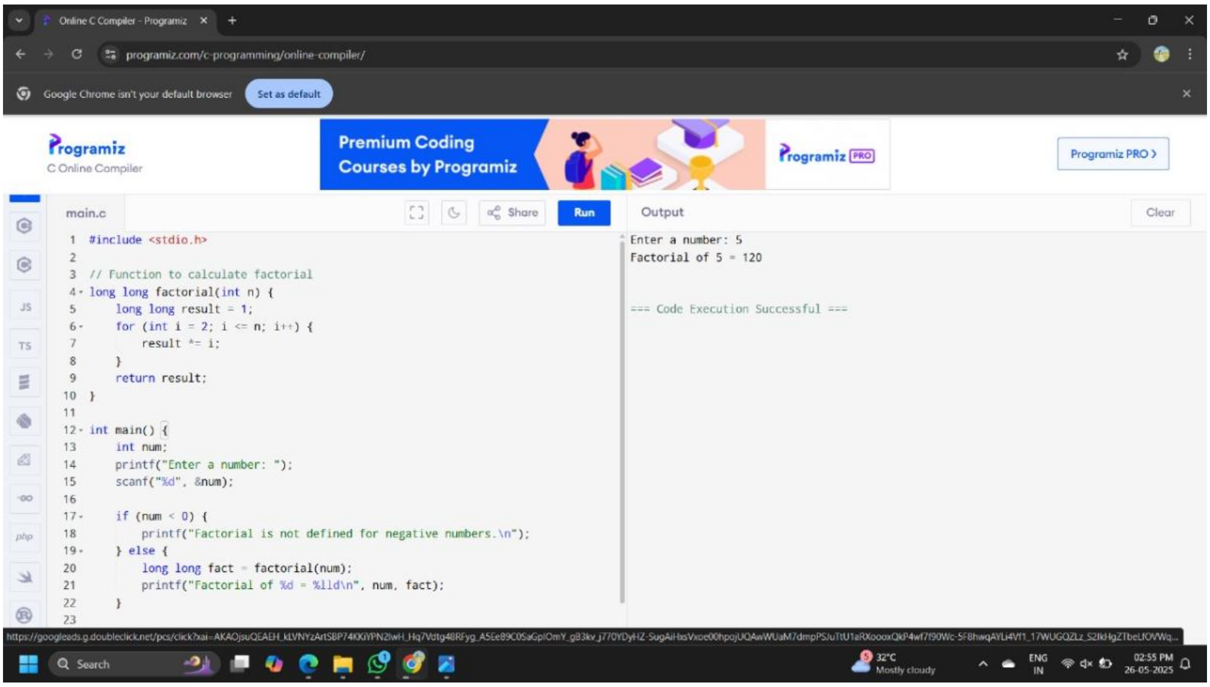
Input:

Enter a number = 5

Output:

Factorial of 5 = 120

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO ›
main.c
Share
Run
Output
Clear

```c
#include <stdio.h>

// Function to calculate factorial
long long factorial(int n) {
    long long result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (num < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        long long fact = factorial(num);
        printf("Factorial of %d = %lld\n", num, fact);
    }

```

```
Enter a number: 5
Factorial of 5 = 120


=== Code Execution Successful ===
```

https://googleads.g.doubleclick.net/pcs/click?xai=AKAOjsuQEAEH_kLVNYzArtSBP74KKiYPN2lwH_Hq7Vdtg48RFyg_A5Ee89C0SaGplOmY_gB3kv_j770YDyHZ-SugAiHxsVxoe00hpojUQAwWUaM7dmpPSJuTtU1aRXoooxQkP4wf7f90Wc-5F8hwqAYLi4Vf1_17WUGQZLz_S2lkHgZTbeLfOVWq...
Search
32°C
Mostly cloudy
ENG
IN
02:55 PM
26-05-2025

4. Write a C program to find Fibonacci series without using Recursion.

Aim:

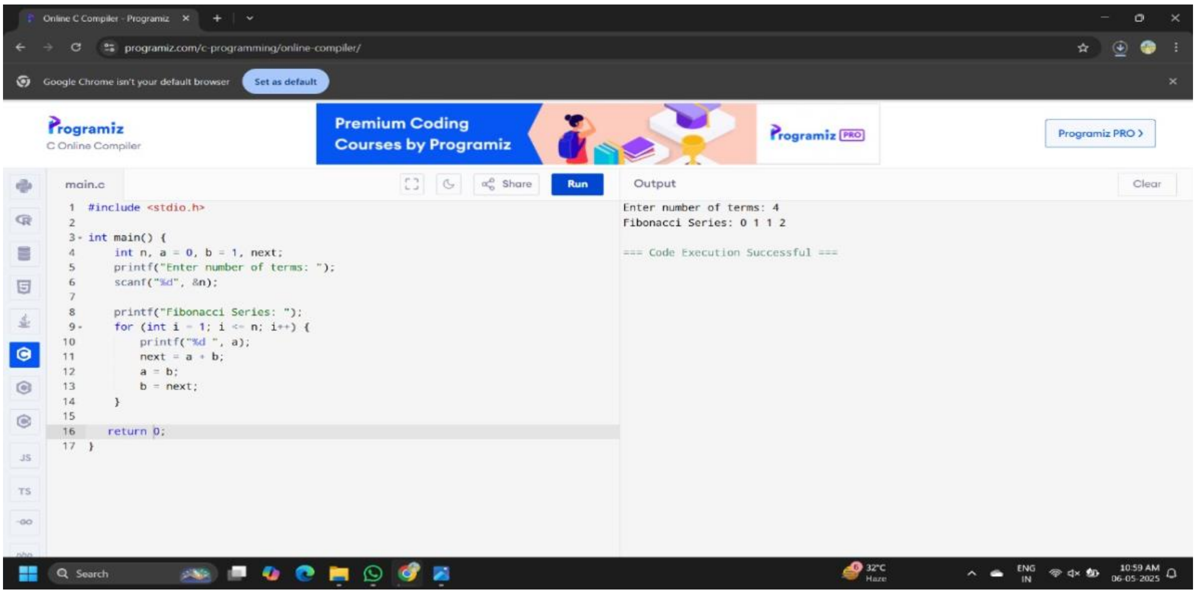
To write a C program to generate the Fibonacci series up to n terms without using recursion.

Algorithm:

```
1. Start.
2. Read the number of terms n.
3. Initialize three variables:
first = 0,
second = 1,
next.
4. Print first and second.
5. Repeat steps 6–7 for i from 3 to n:
6. Set next = first + second
7. Print next.
8. Update first = second, second = next.
9. Stop.
```

Input: Enter the number of terms : 4

Output : Fibonacci Series: 0 1 1 2



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO ›
main.c
Share
Run
Output
Clear

```c
#include <stdio.h>

int main() {
    int n, a = 0, b = 1, next;
    printf("Enter number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");
    for (int i = 1; i <= n; i++) {
        printf("%d ", a);
        next = a + b;
        a = b;
        b = next;
    }

   return 0;
}
```

```
Enter number of terms: 4
Fibonacci Series: 0 1 1 2

=== Code Execution Successful ===
```

5. Write a C program to find Factorial of a given number using Recursion.

Aim:

To write a C program to find the factorial of a given number using recursion.

Algorithm:

```
1. Start.
2. Define a recursive function factorial(int n):
If n == 0 or n == 1, return 1.
Else return n * factorial(n - 1).
3. In the main() function:
Read an integer number n from the user.
Call the recursive function factorial(n) and store the result.
Display the result.
4. Stop.
```

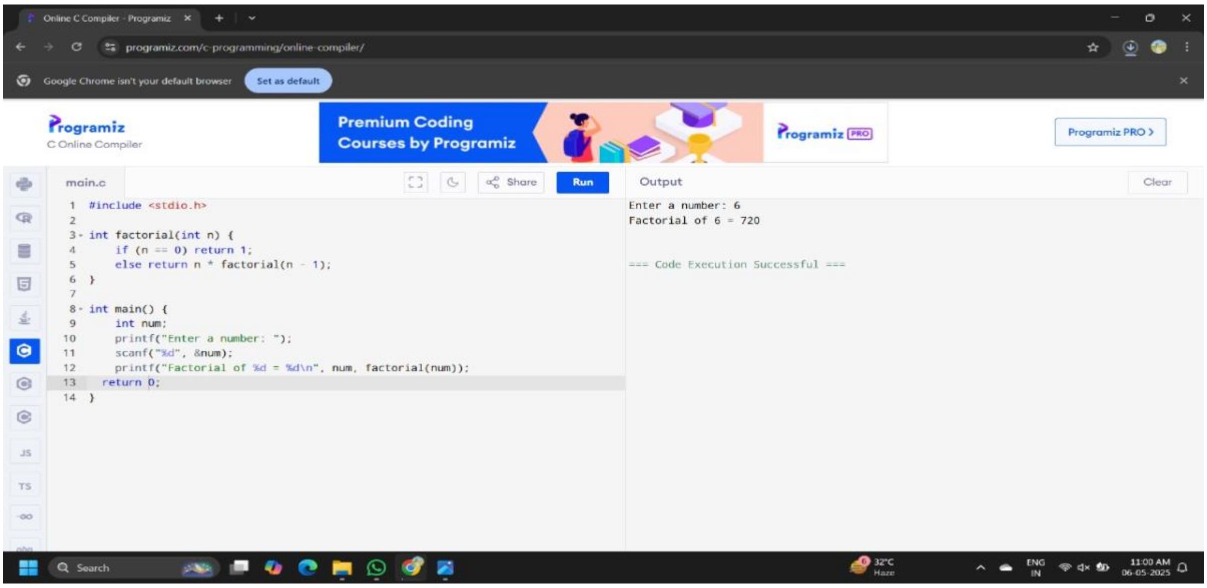
Input:

Enter a Number : 6

Output:

Factorial of 6 = 720

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO >
main.c
Share
Run
Output
Clear

```c
#include <stdio.h>

int factorial(int n) {
    if (n == 0) return 1;
    else return n * factorial(n - 1);
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Factorial of %d = %d\n", num, factorial(num));
  return 0;
}
```

```
Enter a number: 6
Factorial of 6 = 720


=== Code Execution Successful ===
```

6. Write a C program to find Fibonacci series using Recursion.

Aim:

To write a C program to generate the Fibonacci series up to n terms using recursion.

Algorithm:

1. Start.

2. Define a recursive function fibonacci(int n):

If n == 0, return 0.

If n == 1, return 1.

Else return fibonacci(n - 1) + fibonacci(n - 2).

3. In the main() function:

Read the number of terms n.

Loop from 0 to n - 1:

Call fibonacci(i) and print the result.

4. Stop.

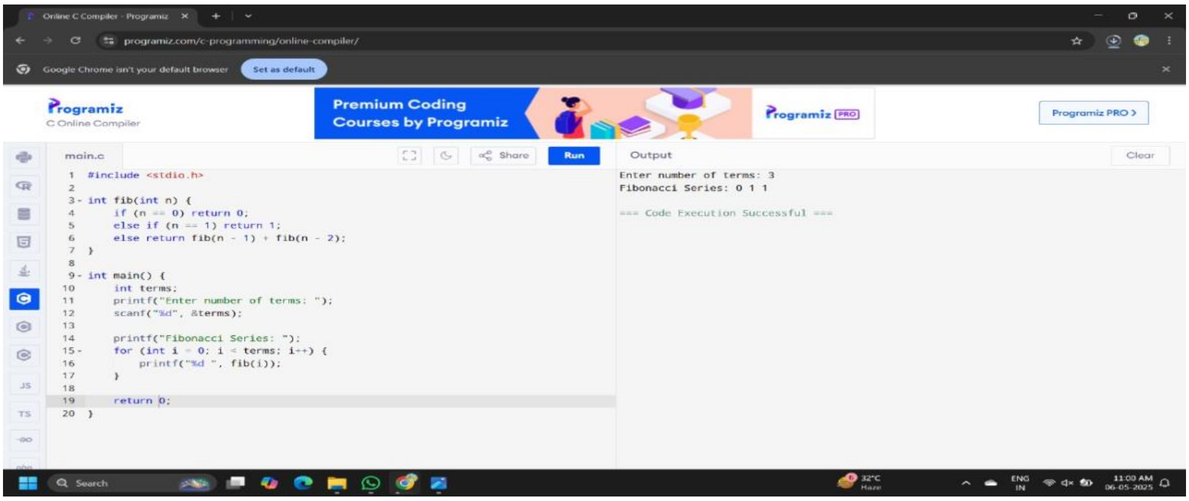
Input:

Enter the number of terms: 3

Output:

Fibonacci Series: 0 1 1

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO >
main.c
Share
Run
Output
Clear
Search

```c
#include <stdio.h>

int fib(int n) {
    if (n == 0) return 0;
    else if (n == 1) return 1;
    else return fib(n - 1) + fib(n - 2);
}

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

    printf("Fibonacci Series: ");
    for (int i = 0; i < terms; i++) {
        printf("%d ", fib(i));
    }

    return 0;
}
```

```
Enter number of terms: 3
Fibonacci Series: 0 1 1

=== Code Execution Successful ===
```

7. Write a C program to implement Array operations such as Insert, Delete and

Display.

Aim:

To write a C program that performs basic array operations: insertion of an element, deletion of an element, and displaying the array elements.

Algorithm:

1. Start

2. Declare an array and variables for size and position.

3. Display a menu with choices: Insert, Delete, Display, and Exit.

4. For Insert:

Input the element and the position.

Shift elements to the right from the position to the end.

Insert the element.

Increment size.

5. For Delete:

Input the position.

Shift elements to the left from the position to the end.

Decrease size.

6. For Display:

Print all elements up to current size.

7. Repeat until Exit is selected.

8. End

Input: 1. Insert

2. delete

3. display

4.Exit

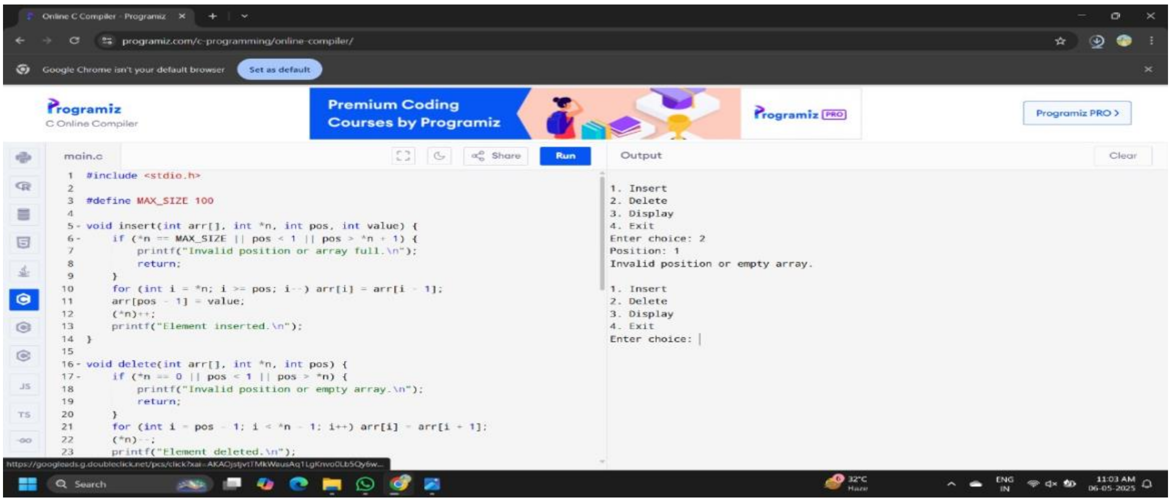
Enter choice : 2

Position : 1

Output:

Invalid position or empty array

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO >
main.c
Share
Run
Output
Clear

```c
#include <stdio.h>

#define MAX_SIZE 100

void insert(int arr[], int *n, int pos, int value) {
    if (*n == MAX_SIZE || pos < 1 || pos > *n + 1) {
        printf("Invalid position or array full.\n");
        return;
    }
    for (int i = *n; i >= pos; i--) arr[i] = arr[i - 1];
    arr[pos - 1] = value;
    (*n)++;
    printf("Element inserted.\n");
}

void delete(int arr[], int *n, int pos) {
    if (*n == 0 || pos < 1 || pos > *n) {
        printf("Invalid position or empty array.\n");
        return;
    }
    for (int i = pos - 1; i < *n - 1; i++) arr[i] = arr[i + 1];
    (*n)--;
    printf("Element deleted.\n");
```

```
1. Insert
2. Delete
3. Display
4. Exit
Enter choice: 2
Position: 1
Invalid position or empty array.

1. Insert
2. Delete
3. Display
4. Exit
Enter choice:
```

8. Write a C program to search a number using Linear Search method.

Aim

To search for a given number in an array using the Linear Search method.

Algorithm

1. Start from the first element of the array.

2. Compare the target element with each element of the array.

3. If the target element is found, return its index.

4. If the target element is not found after traversing the entire array, return -1 (or a suitable indicator).

Input:

Enter no of elements in array : 3

Enter three integers: 1 2 4

Enter the no to search : 4

Output:

4 found at position 3

9. Write a C program to search a number using Binary Search method.

Aim

To search for a given number in a sorted array using the Binary Search method.

Algorithm

1. Find the middle element of the array.

2. Compare the target element with the middle element.

3. If the target element is equal to the middle element, return its index.

4. If the target element is less than the middle element, repeat the process with the left half of the array.

5. If the target element is greater than the middle element, repeat the process with the right half of the array.

6. If the target element is not found after the search space is empty, return -1 (or a suitable indicator).

Input:

Enter number of elements in array: 2

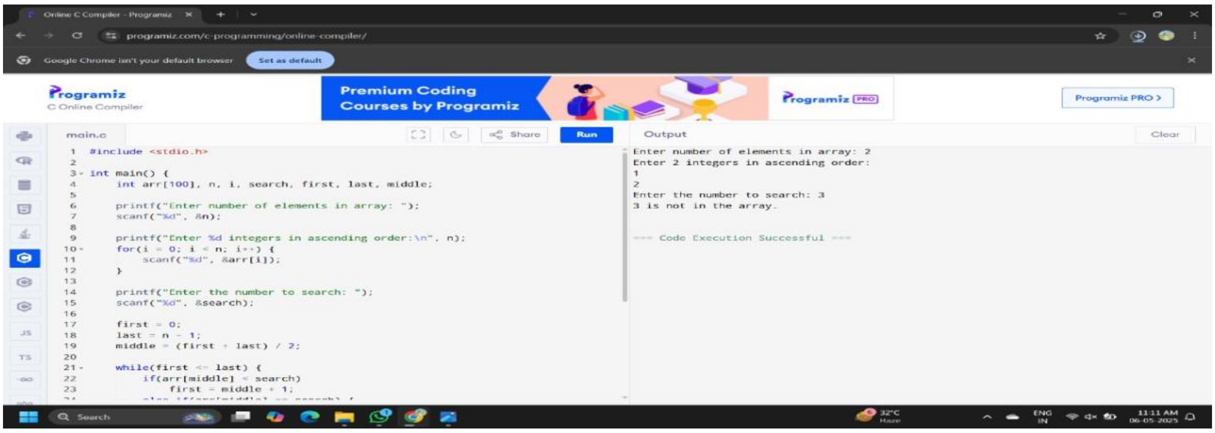
Enter 2 integers in ascending order: 1 2

Enter the no to search : 3

Output:

3 is not in the array

Code:



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO >
main.c
Share
Run
Output
Clear
Search

```c
#include <stdio.h>

int main() {
    int arr[100], n, i, search, first, last, middle;

    printf("Enter number of elements in array: ");
    scanf("%d", &n);

    printf("Enter %d integers in ascending order:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

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

    first = 0;
    last = n - 1;
    middle = (first + last) / 2;

    while(first <= last) {
        if(arr[middle] < search)
            first = middle + 1;
```

```
Enter number of elements in array: 2
Enter 2 integers in ascending order:
1
2
Enter the number to search: 3
3 is not in the array.


=== Code Execution Successful ===
```

10. Write a C program to implement Linked list operations.

Aim

To implement basic operations on a singly linked list, including insertion, deletion, and traversal.

Algorithm

1. Node structure: Define a node structure with an integer data field and a pointer to the next node.

2. Insertion:

   - Insert at the beginning: Create a new node and update the head pointer.

   - Insert at the end: Traverse the list to find the last node and append a new node.

3. Deletion:

   - Delete from the beginning: Update the head pointer to the next node.

   - Delete from the end: Traverse the list to find the second last node and update its next pointer to NULL.

4. Traversal: Traverse the list and print the data of each node.

Input :

1. Insert

2. delete

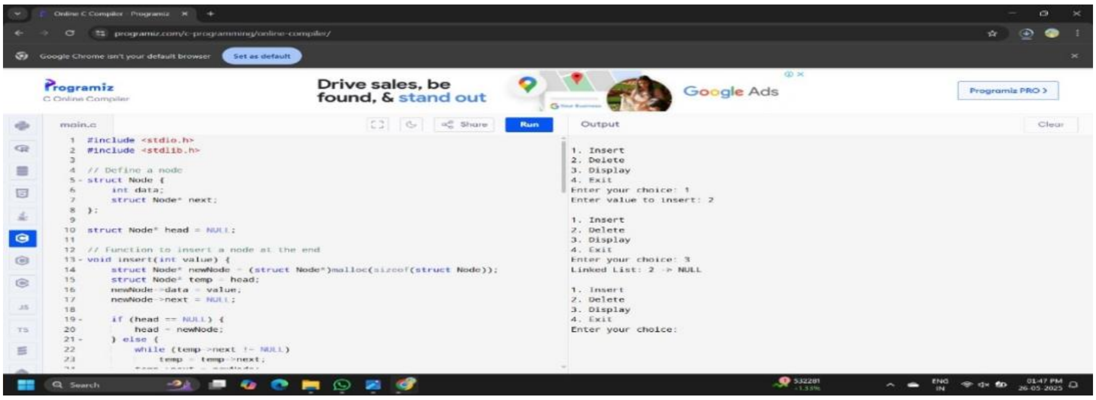
3. display

4.Exit

Enter your choice:3

Output:

Linked list : 2 > null



Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Google Chrome isn't your default browser
Set as default
Programiz
C Online Compiler
Programiz PRO >
main.c
Share
Run
Output
Clear
Search

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

// Define a node
struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

// Function to insert a node at the end
void insert(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    struct Node* temp = head;
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
    } else {
        while (temp->next != NULL)
            temp = temp->next;
```

```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 2

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Linked List: 2 -> NULL

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
```