

EX:1

```
1 #include <stdio.h>
2
3 int main() {
4     int n, rev = 0, rem;
5     printf("Enter a 32-bit integer: ");
6     scanf("%d", &n);
7
8     while(n != 0) {
9         rem = n % 10;
10        rev = rev * 10 + rem;
11        n /= 10;
12    }
13
14    printf("Reversed number = %d\n", rev);
15    return 0;
16 }
```

Enter a 32-bit integer: 1234
Reversed number = 4321

=== Code Execution Successful ===

EX: 2

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

int main() {
    char str[100];
    int i, valid = 1;

    printf("Enter a string: ");
    gets(str);

    for(i = 0; str[i] != '\0'; i++) {
        if(!isalpha(str[i])) {
            valid = 0;
            break;
        }
    }
}
```

Enter a 32-bit integer: 1234
Reversed number = 4321

=== Code Execution Successful ===

```
9     gets(str);
10
11     for(i = 0; str[i] != '\0'; i++) {
12         if(!isalpha(str[i])) {
13             valid = 0;
14             break;
15         }
16     }
17
18     if(valid)
19         printf("Valid String (only alphabets)\n");
20     else
21         printf("Invalid String\n");
22     return 0;
23 }
24 |
```

Enter a 32-bit integer: 1234
Reversed number = 4321

=== Code Execution Successful ===

EX:6 A

```
1 #include <stdio.h>
2
3 int main() {
4     int n, key, i, found = 0;
5
6     printf("Enter number of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter %d elements:\n", n);
11    for(i = 0; i < n; i++)
12        scanf("%d", &arr[i]);
13
14    printf("Enter element to search: ");
15    scanf("%d", &key);
16
17    for(i = 0; i < n; i++) {
18        if(arr[i] == key) {
19            printf("Element %d found at position %d (index %d)\n", key, i
20                + 1, i);
21            found = 1;
22            break;
23        }
24    }
25    if(!found)
26        printf("Element %d not found in array\n", key);
27
28    return 0;
29 }
30
```

Enter number of elements: 3
Enter 3 elements:
10 20 40
Enter element to search: 20
Element 20 found at position 2 (index 1)

=== Code Execution Successful ===

Enter number of elements: 3
Enter 3 elements:
10 20 40
Enter element to search: 20
Element 20 found at position 2 (index 1)

=== Code Execution Successful ===

EX 6 B

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int n, key, low, high, mid, found = 0;
5
6     printf("Enter number of elements: ");
7     scanf("%d", &n);
8
9     int arr[n];
10    printf("Enter %d elements in sorted order:\n", n);
11    for(int i = 0; i < n; i++)
12        scanf("%d", &arr[i]);
13
14    printf("Enter element to search: ");
15    scanf("%d", &key);
16
17    low = 0;
18    high = n - 1;
19
20    while(low <= high) {
21        mid = (low + high) / 2;
22
23        if(arr[mid] == key) {
24            printf("Element %d found at position %d (index %d)\n", key,
25                mid + 1, mid);
26            found = 1;
27            break;
28        }
29        else if(arr[mid] < key)
30            low = mid + 1;
31        else
32            high = mid - 1;
33    }
34
35    if(!found)
36        printf("Element %d not found in array\n", key);
37
38    return 0;
39 }
```

Enter number of elements: 3
Enter 3 elements in sorted order:
10 20 30
Enter element to search: 30
Element 30 found at position 3 (index 2)

=== Code Execution Successful ===

Enter number of elements: 3
Enter 3 elements in sorted order:
10 20 30
Enter element to search: 30
Element 30 found at position 3 (index 2)

=== Code Execution Successful ===

```
26         break;
27     }
28     else if(arr[mid] < key)
29         low = mid + 1;
30     else
31         high = mid - 1;
32 }
33
34 if(!found)
35     printf("Element %d not found in array\n", key);
36
37 return 0;
38 }
```

Enter number of elements: 3
Enter 3 elements in sorted order:
10 20 30
Enter element to search: 30
Element 30 found at position 3 (index 2)

=== Code Execution Successful ===

EX:7

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {10, 20, 30, 40, 50}, key, i, found = 0;
5     printf("Enter element to find: ");
6     scanf("%d", &key);
7
8     for(i = 0; i < 5; i++) {
9         if(arr[i] == key) {
10             printf("Element found at index %d\n", i);
11             found = 1;
12             break;
13         }
14     }
15     if(!found)
16         printf("Element not found\n");
17     return 0;
}
```

Enter element to find: 20
Element found at index 1

=== Code Execution Successful ===

EX:8

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {2, 5, 8, 11, 14};
5     printf("Even numbers: ");
6     for(int i=0; i<5; i++)
7         if(arr[i] % 2 == 0)
8             printf("%d ", arr[i]);
9
10    printf("\nOdd numbers: ");
11    for(int i=0; i<5; i++)
12        if(arr[i] % 2 != 0)
13            printf("%d ", arr[i]);
14
15    return 0;
16 }
```

Even numbers: 2 8 14
Odd numbers: 5 11

=== Code Execution Successful ===

EX:9

```
1 #include <stdio.h>
2
3 int main() {
4     int n, a=0, b=1, c, sum=1;
5     printf("Enter terms: ");
6     scanf("%d", &n);
7
8     for(int i=2; i<n; i++) {
9         c = a + b;
10        sum += c;
11        a = b;
12        b = c;
13    }
14    printf("Sum of Fibonacci series = %d\n", sum);
15    return 0;
16 }
```

Enter terms: 5
Sum of Fibonacci series = 7

=== Code Execution Successful ===

EX:10

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int n, fact = 1;
5     printf("Enter number: ");
6     scanf("%d", &n);
7
8     for(int i=1; i<=n; i++)
9         fact *= i;
10
11    printf("Factorial = %d\n", fact);
12    return 0;
13 }
14
15
```

Output

Enter number: 5
Factorial = 120

=== Code Execution Successful ===

EX:17

```
1 #include <stdio.h>
2
3 int main() {
4     int arr[5] = {10, 20, 30, 40, 50};
5     int key, choice, i, found = 0;
6     int low, high, mid;
7
8     printf("Array elements: 10 20 30 40 50\n");
9     printf("\nChoose search method:\n");
10    printf("1. Linear Search\n");
11    printf("2. Binary Search\n");
12    printf("Enter your choice: ");
13    scanf("%d", &choice);
14
15    printf("Enter number to search: ");
16    scanf("%d", &key);
```

Array elements: 10 20 30 40 50

Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

=== Code Execution Successful ===

```
17
18~ switch(choice) {
19     case 1: // Linear Search
20~         for(i = 0; i < 5; i++) {
21~             if(arr[i] == key) {
22                 printf("Element found at position %d (index %d)\n", i
                    + 1, i);
23                 found = 1;
24                 break;
25             }
26         }
27         if(!found)
28             printf("Element not found\n");
29         break;
30
31     case 2: // Binary Search
32         low = 0;
33         high = 4;
34         found = 0;
35
36~         while(low <= high) {
37             mid = (low + high) / 2;
38
39~             if(arr[mid] == key) {
40                 printf("Element found at position %d (index %d)\n",
                    mid + 1, mid);
41                 found = 1;
42                 break;
43             } else if(arr[mid] < key)
44                 low = mid + 1;
45             else
46                 high = mid - 1;
47         }
48
49         if(!found)
50             printf("Element not found\n");
51         break;
52
53     default:
54         printf("Invalid choice! Please enter 1 or 2.\n");
55 }
56
57 return 0;
58 }
59
```

Array elements: 10 20 30 40 50

Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

=== Code Execution Successful ===

main.c

Array elements: 10 20 30 40 50

Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

=== Code Execution Successful ===

Array elements: 10 20 30 40 50

Choose search method:
1. Linear Search
2. Binary Search
Enter your choice: 1
Enter number to search: 10
Element found at position 1 (index 0)

=== Code Execution Successful ===

EX:31

```
1 #include <stdio.h>
2 #include <string.h>
3
4~ int main() {
5     char str[100];
6     int i, j, len;
7
8     printf("Enter a string: ");
9     fgets(str, sizeof(str), stdin); // Safe input function
10
11     // Remove newline character added by fgets
12     str[strcspn(str, "\n")] = '\0';
13
14     len = strlen(str);
15     printf("\nRepeated characters and their indexes:\n");
16
```

Enter a string: HELLO

Repeated characters and their indexes:
Character 'L' repeated at index 2 and 3

=== Code Execution Successful ===

```
16
17-   for(i = 0; i < len; i++) {
18-       for(j = i + 1; j < len; j++) {
19-           if(str[i] == str[j] && str[i] != ' ') {
20-               printf("Character '%c' repeated at index %d and %d\n",
21-                   str[i], i, j);
22-           }
23-       }
24-   }
25-   return 0;
26- }
27
28
29
```

Enter a string: HELLO

Repeated characters and their indexes:
Character 'L' repeated at index 2 and 3

=== Code Execution Successful ===

EX:32

```
1  #include <stdio.h>
2
3- int main() {
4-     int arr[] = {1, 2, 2, 3, 3, 3, 4};
5-     int n = 7, count, maxCount = 0, freqNum;
6
7-     for(int i=0; i<n; i++) {
8-         count = 1;
9-         for(int j=i+1; j<n; j++) {
10-             if(arr[i] == arr[j])
11-                 count++;
12-         }
13-         if(count > maxCount) {
14-             maxCount = count;
15-             freqNum = arr[i];
16-         }
17
18-     }
19-     printf("Most frequent number: %d (count = %d)\n", freqNum, maxCount);
20-     return 0;
21- }
```

Most frequent number: 3 (count = 3)

=== Code Execution Successful ===

```
9-     for(int j=i+1; j<n; j++) {
10-         if(arr[i] == arr[j])
11-             count++;
12-     }
13-     if(count > maxCount) {
14-         maxCount = count;
15-         freqNum = arr[i];
16-     }
17- }
18
19- printf("Most frequent number: %d (count = %d)\n", freqNum, maxCount);
20- return 0;
21- }
```

Most frequent number: 3 (count = 3)

=== Code Execution Successful ===

EX:42

```
main.c
1  #include <stdio.h>
2
3- int main() {
4-     int a[2][2] = {{1, 2}, {3, 4}};
5-     int rowSum, colSum;
6
7-     for(int i=0; i<2; i++) {
8-         rowSum = 0;
9-         for(int j=0; j<2; j++)
10-             rowSum += a[i][j];
11-         printf("Sum of row %d = %d\n", i, rowSum);
12-     }
13
14-     for(int j=0; j<2; j++) {
15-         colSum = 0;
16-         for(int i=0; i<2; i++)
```

Sum of row 0 = 3
Sum of row 1 = 7
Sum of column 0 = 4
Sum of column 1 = 6

=== Code Execution Successful ===

```
11     printf("Sum of row %d = %d\n", i, rowSum);
12 }
13
14 for(int j=0; j<2; j++) {
15     colSum = 0;
16     for(int i=0; i<2; i++)
17         colSum += a[i][j];
18     printf("Sum of column %d = %d\n", j, colSum);
19 }
20
21 return 0;
22 }
23
```

Sum of row 0 = 3
Sum of row 1 = 7
Sum of column 0 = 4
Sum of column 1 = 6

=== Code Execution Successful ===

EX:43

```
main.c
3 int main() {
4     int arr[] = {1, 2, 2, 3, 4, 4};
5     int n = 6, i, j;
6
7     printf("Elements repeated twice: ");
8     for(i=0; i<n; i++) {
9         int count = 0;
10        for(j=0; j<n; j++) {
11            if(arr[i] == arr[j])
12                count++;
13        }
14        if(count == 2)
15            printf("%d ", arr[i]);
16    }
17    return 0;
18 }
```

Sum of row 0 = 3
Sum of row 1 = 7
Sum of column 0 = 4
Sum of column 1 = 6

=== Code Execution Successful ===

EX:48

```
2
3 int main() {
4     int a[3][3] = {
5         {1, 9, 3},
6         {4, 2, 8},
7         {7, 6, 5}
8     };
9     int max = a[0][0];
10
11     for(int i=0; i<3; i++)
12         for(int j=0; j<3; j++)
13             if(a[i][j] > max)
14                 max = a[i][j];
15
16     printf("Largest element in matrix = %d\n", max);
17     return 0;
18 }
```

Largest element in matrix = 9

=== Code Execution Successful ===

EX:

```
1 #include <stdio.h>
2
3 int main() {
4     int a[2][2], b[2][2], c[2][2];
5     int i, j, k, choice;
6
7     printf("Enter elements of Matrix A (2x2):\n");
8     for(i=0;i<2;i++){
9         for(j=0;j<2;j++){
10             scanf("%d",&a[i][j]);
11         }
12     }
13     printf("Enter elements of Matrix B (2x2):\n");
14     for(i=0;i<2;i++){
15         for(j=0;j<2;j++){
16             scanf("%d",&b[i][j]);
```

```
5 4
Enter elements of Matrix B (2x2):
2 3
4 5

1.Addition
2.Subtraction
3.Multiplication
4.Traversal
Enter choice: 2

Subtraction:
-1 -1
1 -1
```

```
17     printf("\n1.Addition\n2.Subtraction\n3.Multiplication\n4
    .Traversal\nEnter choice: ");
18     scanf("%d",&choice);
19
20     if(choice==1){
21         printf("\nAddition:\n");
22         for(i=0;i<2;i++){
23             for(j=0;j<2;j++){
24                 c[i][j]=a[i][j]+b[i][j];
25                 printf("%d ",c[i][j]);
26             }
27             printf("\n");
28         }
29     }
30     else if(choice==2){
31         printf("\nSubtraction:\n");
```

```
5 4
Enter elements of Matrix B (2x2):
2 3
4 5

1.Addition
2.Subtraction
3.Multiplication
4.Traversal
Enter choice: 2

Subtraction:
-1 -1
1 -1
```

```
32     for(i=0;i<2;i++){
33         for(j=0;j<2;j++){
34             c[i][j]=a[i][j]-b[i][j];
35             printf("%d ",c[i][j]);
36         }
37         printf("\n");
38     }
39 }
40 else if(choice==3){
41     printf("\nMultiplication:\n");
42     for(i=0;i<2;i++){
43         for(j=0;j<2;j++){
44             c[i][j]=0;
45             for(k=0;k<2;k++){
46                 c[i][j]+=a[i][k]*b[k][j];
47             }
48             printf("%d ",c[i][j]);
```

```
5 4
Enter elements of Matrix B (2x2):
2 3
4 5

1.Addition
2.Subtraction
3.Multiplication
4.Traversal
Enter choice: 2

Subtraction:
-1 -1
1 -1
```

```
48     }
49     printf("\n");
50 }
51 }
52 else if(choice==4){
53     printf("\nMatrix A:\n");
54     for(i=0;i<2;i++){
55         for(j=0;j<2;j++){
56             printf("%d ",a[i][j]);
57         }
58     }
59
60     printf("\nMatrix B:\n");
61     for(i=0;i<2;i++){
62         for(j=0;j<2;j++){
63             printf("%d ",b[i][j]);
```

```
5 4
Enter elements of Matrix B (2x2):
2 3
4 5

1.Addition
2.Subtraction
3.Multiplication
4.Traversal
Enter choice: 2

Subtraction:
-1 -1
1 -1
```



```
61-         for(i=0;i<2;i++){
62-             for(j=0;j<2;j++){
63-                 printf("%d ",b[i][j]);
64-                 printf("\n");
65-             }
66-         }
67-     else
68-         printf("Invalid choice!");
69-
70-     return 0;
71- }
72-
73-
74-
75-
76-
```

```
5 4
Enter elements of Matrix B (2x2):
2 3
4 5

1.Addition
2.Subtraction
3.Multiplication
4.Traversal
Enter choice: 2

Subtraction:
-1 -1
1 -1
```