

## 1. Write a C program to declare, initialize, input elements in array and print array.

```
1 //program to print array elements by youtube:  
2 #include<stdio.h>  
3 int main()  
4 {  
5     int i,a[10];  
6     printf("Enter the array elements:\n");  
7     for(i=0;i<10;i++)  
8     {  
9         scanf("%d",&a[i]);  
10    }  
11    printf("the entered array elements are:\n");  
12    for(i=0;i<10;i++)  
13    {  
14        printf("%d ",a[i]);  
15    }  
16    return 0;  
17 }
```

/tmp/cqxEev72Ny.o  
Enter the array elements:  
1  
12  
123  
159  
156  
4567  
8  
7  
64  
2  
the entered array elements are:  
1 12 123 159 156 4567 8 7 64 2  
== Code Execution Successful ==

```
1 //program to print array elements by codeforwin:  
2 #include<stdio.h>  
3 #define MAX_SIZE 1000  
4 int main()  
5 {  
6     int n,i;  
7     int arr[MAX_SIZE];  
8     printf("enter the size of the array: ");  
9     scanf("%d",&n);  
10    printf("enter the elements of the array:\n");  
11    for(i=0;i<n;i++)  
12    {  
13        scanf("%d",&arr[i]);  
14    }  
15    printf("the elements that entered are: ");  
16    for(i=0;i<n;i++)  
17    {  
18        printf("%d ",arr[i]);  
19    }  
20    return 0;  
21 }
```

/tmp/71c56f507W.o  
enter the size of the array: 10  
enter the elements of the array:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
the elements that entered are: 1 2 3 4 5 6 7 8 9 10  
== Code Execution Successful ==

## 2. Write a C program to print all negative and positive elements in an array.

```
main.c  
1 //print positive and negative elements in an array  
2 #include<stdio.h>  
3 #define MAX_SIZE 1000  
4 int main()  
5 {  
6     int n,i;  
7     int arr[MAX_SIZE];  
8     printf("Enter the size of the array:");  
9     scanf("%d",&n);  
10    printf("Enter the array elements:");  
11    for(i=0;i<n;i++)  
12    {  
13        scanf("%d",&arr[i]);  
14    }  
15    printf("The positive elements are:");  
16    for(i=0;i<n;i++)  
17    {  
18        if(arr[i]>0)  
19        {  
20            printf("%d ",arr[i]);  
21        }  
22    }  
23    printf("\nThe negative elements are:");  
24    for(i=0;i<n;i++)  
25    {  
26        if(arr[i]<0)  
27        {  
28            printf("%d ",arr[i]);  
29        }  
30    }  
31    return 0;  
32 }
```

/tmp/u54paauz3w7.o  
Enter the size of the array:15  
Enter the array elements:5 8 9 4 -8 6 -9 4 9 -1 -2 -3 -7 -4 8  
The positive elements are:5 8 9 4 6 4 9 8  
the negative elements are:-8 -9 -1 -2 -3 -7 -4  
== Code Execution Successful ==

### 3. Write a C program to find the sum of all array elements.

The screenshot shows a code editor interface with a dark theme. On the left, the code file 'main.c' is displayed with the following content:

```
1 //sum aof array elements
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,sum=0;
7     int arr[MAX_SIZE];
8     printf("Enter the size of the array:");
9     scanf("%d",&n);
10    printf("Enter the array elements:");
11    for(i=0;i<n;i++)
12    {
13        scanf("%d",&arr[i]);
14        sum=sum+arr[i];
15    }
16    printf("The sum is:%d",sum);
17    return 0;
18 }
19
```

At the top of the editor, there are several icons: a file icon, a lightbulb icon, a share icon, and a 'Run' button. To the right of the code area is an 'Output' window containing the execution results:

```
/tmp/6Moe5VoT1Y.o
Enter the size of the array:5
Enter the array elements:50 10 10 10 10
The sum is:90
== Code Execution Successful ==
```

A 'Clear' button is located at the top right of the output window.

### 4. Program to find maximum and minimum element in array.

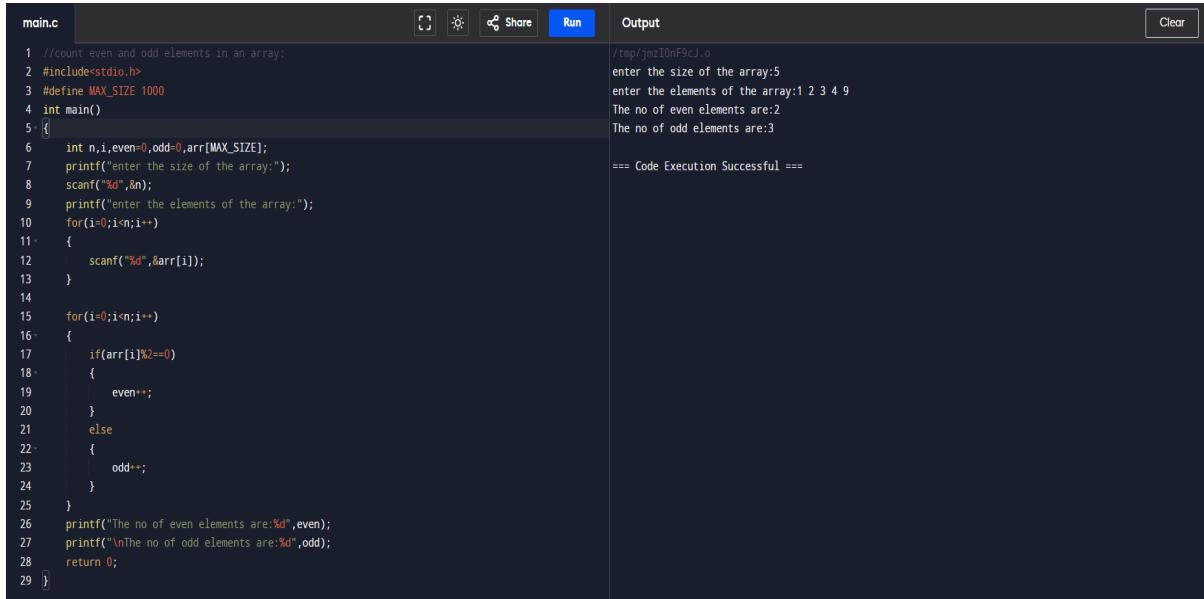
The screenshot shows a code editor interface with a dark theme. On the left, the code file 'main.c' is displayed with the following content:

```
1 #include<stdio.h>
2 #define MAX_SIZE 1000
3 int main()
4 {
5     int i,n,max,min,arr[MAX_SIZE];
6     printf("Enter the size of the array:");
7     scanf("%d",&n);
8     printf("Enter elements in the array:");
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    max=arr[0];
14    min=arr[0];
15    for(i=0;i<n;i++)
16    {
17        if(arr[i]>max)
18        {
19            max=arr[i];
20        }
21        if(arr[i]<min)
22        {
23            min=arr[i];
24        }
25    }
26    printf("Maximum element=%d",max);
27    printf("\nMinimum element=%d",min);
28    return 0;
29 }
```

At the top of the editor, there are several icons: a file icon, a lightbulb icon, a share icon, and a 'Run' button. To the right of the code area is an 'Output' window containing the execution results:

```
/tmp/WkEhJKH3wC.o
Enter the size of the array:10
Enter elements in the array:-10 10 0 20 -2 50 100 20 -1 10
Maximum element=100
Minimum element=-10
== Code Execution Successful ==
```

### 5. Program to count even and odd elements in an array.



```

main.c
1 //count even and odd elements in an array:
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int n,i,even=0,odd=0,arr[MAX_SIZE];
7     printf("enter the size of the array:");
8     scanf("%d",&n);
9     printf("enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]%2==0)
17        {
18            even++;
19        }
20        else
21        {
22            odd++;
23        }
24    }
25    printf("The no of even elements are:%d",even);
26    printf("\nThe no of odd elements are:%d",odd);
27    return 0;
28 }

```

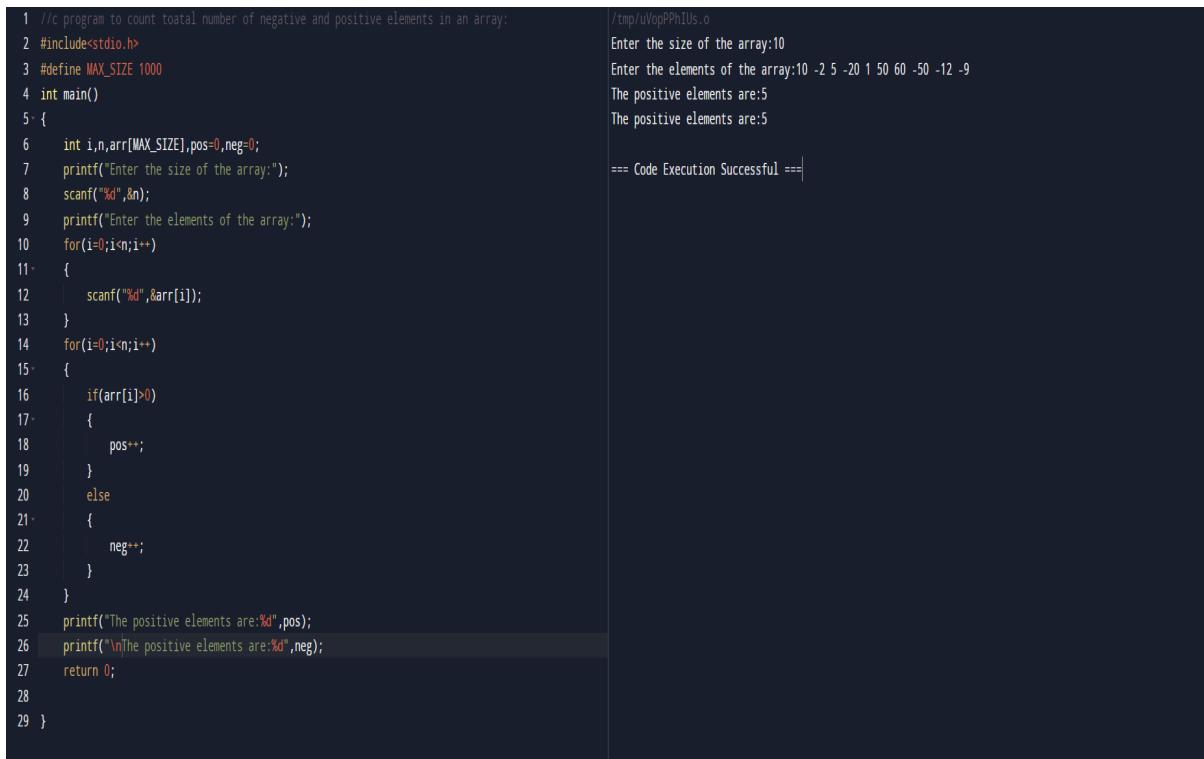
Output

```

/tmp/jmzIOnf9cJ.o
enter the size of the array:5
enter the elements of the array:1 2 3 4 9
The no of even elements are:2
The no of odd elements are:3
*** Code Execution Successful ***

```

## 6. Write a C program to count the total number of negative and positive elements in an array.



```

1 //c program to count total number of negative and positive elements in an array:
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],pos=0,neg=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]>0)
17        {
18            pos++;
19        }
20        else
21        {
22            neg++;
23        }
24    }
25    printf("The positive elements are:%d",pos);
26    printf("\nThe positive elements are:%d",neg);
27    return 0;
28 }

```

Output

```

/tmp/dVopPPhIUo.o
Enter the size of the array:10
Enter the elements of the array:10 -2 5 -20 1 50 60 -50 -12 -9
The positive elements are:5
The positive elements are:5
*** Code Execution Successful ***

```

## 7.Program to copy array elements to another array:

```
main.c
1 //Program to copy array elements to another array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,src[MAX_SIZE],dest[MAX_SIZE];
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("enter the elements of the array: ");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&src[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        dest[i]=src[i];
17    }
18    printf("The src elements are:");
19    for(i=0;i<n;i++)
20    {
21        printf("%d\t",src[i]);
22    }
23    printf("\nThe dst elements are:");
24    for(i=0;i<n;i++)
25    {
26        printf("%d\t",dest[i]);
27    }
28    return 0;
29 }
30
```

Output

```
/tmp/OrgHLwyzrf.o
Enter the size of the array:5
enter the elements of the array: 1 6 7 9 4
The src elements are:1 6 7 9 4
The dst elements are:1 6 7 9 4
== Code Execution Successful ==
```

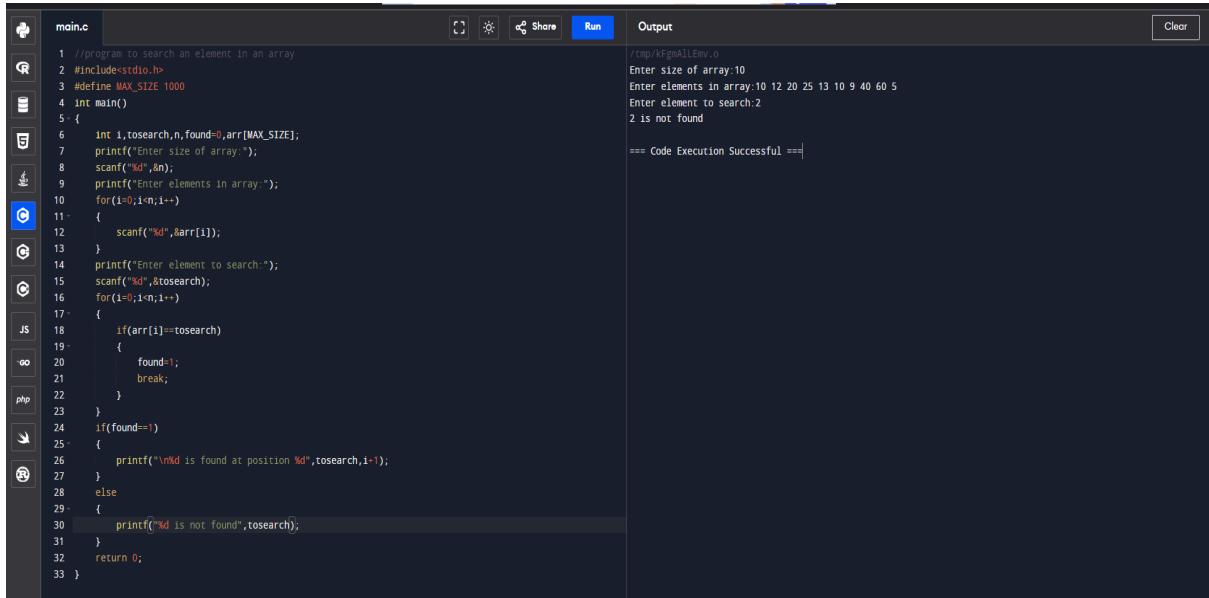
## 8.Program to print array in reverse:

```
main.c
1 //print arary element in reverse
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE];
7     printf("enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the element of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("The reverse of teh array is:");
15    for(i=n-1;i>=0;i--)
16    {
17        printf("%d ",arr[i]);
18    }
19    return 0;
20 }
21
```

Output

```
/tmp/F4GPBPb4Zb.o
enter the size of the array:10
Enter the element of the array:10 20 30 40 50 60 70 80 90 100
The reverse of teh array is:100 90 80 70 60 50 40 30 20 10
== Code Execution Successful ==
```

## 9.Program to search element in array

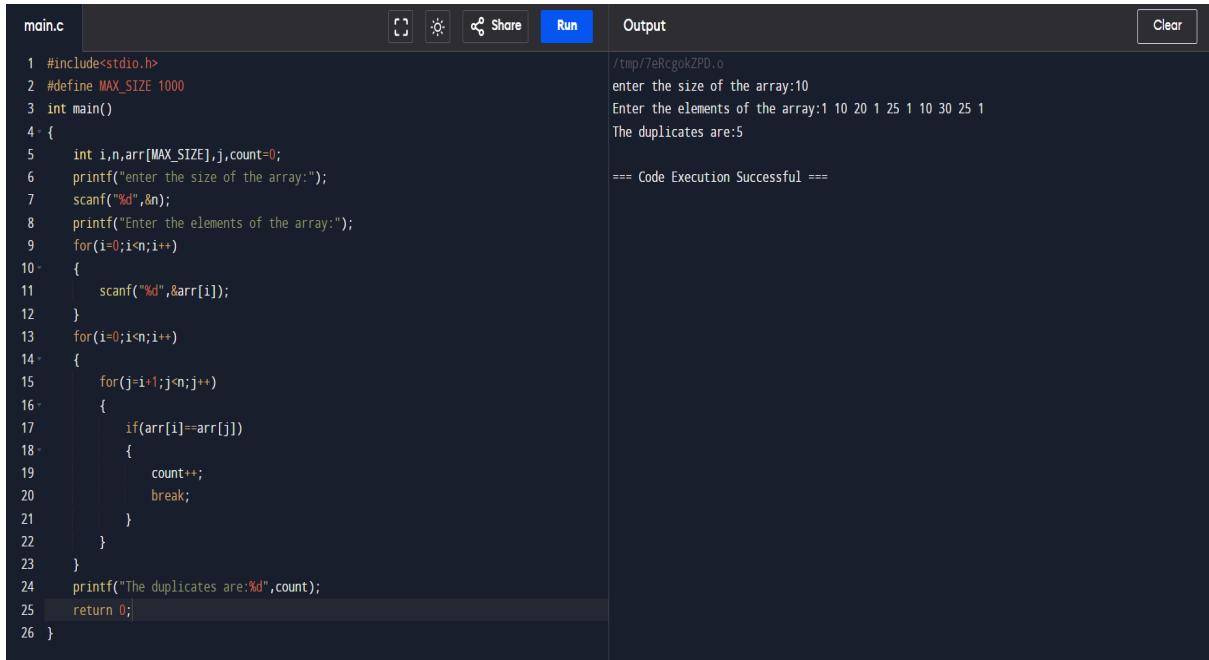


```
main.c
1 //program to search an element in an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,tosearch,n,found=0,arr[MAX_SIZE];
7     printf("Enter size of array:");
8     scanf("%d",&n);
9     printf("Enter elements in array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("Enter element to search:");
15    scanf("%d",&tosearch);
16    for(i=0;i<n;i++)
17    {
18        if(arr[i]==tosearch)
19        {
20            found=1;
21            break;
22        }
23    }
24    if(found==1)
25    {
26        printf("\n%d is found at position %d",tosearch,i+1);
27    }
28    else
29    {
30        printf("%d is not found",tosearch);
31    }
32    return 0;
33 }
```

Output

```
/tmp/KFpmAllFmv.o
Enter size of array:10
Enter elements in array:10 12 20 25 13 10 9 40 60 5
Enter element to search:2
2 is not found
==== Code Execution Successful ===
```

## 10.Program to count duplicate elements in array



```
main.c
1 #include<stdio.h>
2 #define MAX_SIZE 1000
3 int main()
4 {
5     int i,n,arr[MAX_SIZE],j,count=0;
6     printf("enter the size of the array:");
7     scanf("%d",&n);
8     printf("Enter the elements of the array:");
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    for(i=0;i<n;i++)
14    {
15        for(j=i+1;j<n;j++)
16        {
17            if(arr[i]==arr[j])
18            {
19                count++;
20                break;
21            }
22        }
23    }
24    printf("The duplicates are:%d",count);
25    return 0;
26 }
```

Output

```
/tmp/7eRcgokZPD.o
enter the size of the array:10
Enter the elements of the array:1 10 20 1 25 1 10 30 25 1
The duplicates are:5
==== Code Execution Successful ===
```

## 11.Insert an element into an array at a specified position

The screenshot shows a code editor window with the file name "main.c". The code implements an insertion operation on an array. It first asks for the size of the array, then for its elements, and then for the element to be inserted and its insertion position. The code then shifts elements to make space for the new element and inserts it. Finally, it prints the modified array. The output window shows the execution of the program with input values 3, 1, 2, 3, 654, and 0, resulting in the output: "The element 654 is inserted in the position 0: 654 1 2 3".

```
main.c
1 //insert an element in an specified position
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,position,enum,arr[MAX_SIZE],temp[n];
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("Enter the element to be inserted:");
15    scanf("%d",&enum);
16    printf("Enter the element inserted position:");
17    scanf("%d",&position);
18    position--;
19    for(i=n;i>position;i--)
20    {
21        if(i==position)
22        {
23            temp[i]=arr[i];
24        }
25        if(i>position)
26        {
27            temp[i]=arr[i-1];
28        }
29        if(i==position)
30        {
31            temp[i]=enum;
32        }
33    }
34    printf("The element %d is inserted in the position %d.\n",enum,position);
35    for(i=0;i<n;i++)
36    {
37        printf("%d ",temp[i]);
38    }
39    return 0;
40 }
```

Output  
//tmp/redextvR0t.o  
enter the size of the array:3  
Enter the elements of the array:1 2 3  
Enter the element to be inserted:654  
Enter the element inserted position:1  
The element 654 is inserted in the position 0:  
654 1 2 3  
== Code Execution Successful ==

## 12.Number of occurrences of an number

The screenshot shows a code editor window with the file name "main.c". The code counts how many times a target element appears in an array. It first asks for the size of the array, then for its elements, and then for the target element. It then iterates through the array, counting the occurrences of the target element. The output window shows the execution of the program with input values 4, 1, 2, 1, 2, resulting in the output: "The target element occurrences is:2".

```
main.c
1 //number of occurrences of an target element
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],t,count=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("Enter the target element:");
15    scanf("%d",&t);
16    printf("The target element occurrences is:");
17    for(i=0;i<n;i++)
18    {
19        if(arr[i]==t)
20        {
21            count++;
22        }
23    }
24    printf("%d",count);
25    return 0;
26 }
```

Output  
/tmp/anRJj19aM4.o  
Enter the size of the array:4  
Enter the elements:1 2 1 2  
Enter the target element:2  
The target element occurrences is:2  
== Code Execution Successful ==

## 13.Print square of the elements in an array

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing a C program named 'main.c'. The code prompts the user for the size of an array and its elements, then calculates and prints the square of each element. On the right is the output pane, which shows the execution of the program. It starts with the command '/tmp/9dC4Xi3b1I.o', followed by the user input 'Enter the size of the array:2' and 'Enter the elements of the array:7 8'. The program then outputs 'The square is:49 64' and concludes with the message '== Code Execution Successful =='.

```
main.c
1 //print square of the element in an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],sqr;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("The square is:");
15
16    for(i=0;i<n;i++)
17    {
18        sqr=arr[i]*arr[i];
19        printf("%d ",sqr);
20    }
21    return 0;
22 }
23
```

## 14.Difference between maximum and minimum element of an array

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing a C program named 'main.c'. The code prompts the user for the size of an array and its elements, then finds and prints the maximum and minimum elements along with their difference. On the right is the output pane, which shows the execution of the program. It starts with the command '/tmp/dyxj.VOSok3.o', followed by the user input 'enter the size of the array:5' and 'enter the element of the array:10 20 30 40 100'. The program then outputs 'maximum element is:100', 'minimum element is:10', and 'difference:90', concluding with the message '== Code Execution Successful =='.

```
main.c
1 //print square of the element in an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],max,min,diff;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the element of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    max=arr[0];
15    min=arr[0];
16    for(i=0;i<n;i++)
17    {
18        if(arr[i]>max)
19        {
20            max=arr[i];
21        }
22        if(arr[i]<min)
23        {
24            min=arr[i];
25        }
26    }
27    diff=max-min;
28    printf("maximum element is:%d\n",max);
29    printf("minimum element is:%d\n",min);
30    printf("difference:%d",diff);
31    return 0;
32 }
```

## 15.Sum the array after removing duplicate elements

```

main.c
1 //Sum the array after removing duplicate elements
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],j,sum=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]!=-1)
17        {
18            for(j=i+1;j<n;j++)
19            {
20                if(arr[i]==arr[j])
21                {
22                    arr[j]=-1;
23                }
24            }
25            sum=sum+arr[i];
26        }
27    }
28 }
29 printf("The sum is:%d",sum);
30 return 0;
31 }

```

Output

```

/tmp/novUnp73RM.o
Enter the size of the array:8
enter the elements of the array:1 1 3 3 2 5 5
The sum is:11
*** Code Execution Successful ***

```

## 16.program to find sum of negative and positive integers

```

main.c
1 //program to find sum of negative and positive integers
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],pos=0,neg=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]>0)
17        {
18            pos=pos+arr[i];
19        }
20        else
21        {
22            neg=neg+arr[i];
23        }
24    }
25    printf("the sum of positive integers are:%d",pos);
26    printf("\nthe sum of negative integers are:%d",neg);
27    return 0;
28 }

```

Output

```

/tmp/1111oaf7kh9.o
Enter the size of the array:8
Enter the elements of the array:
2
-4
-5
6
0
7
-4
-9
the sum of positive integers are:15
The sum of negative integers are:-22
*** Code Execution Successful ***

```

## 17.Removing even number from an array

```

main.c
1 //removing even numbers from an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE];
7     printf("Enter the array size:");
8     scanf("%d",&n);
9     printf("Enter the array elements:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]%2!=0)
17        {
18            printf("%d ",arr[i]);
19        }
20    }
21    return 0;
22 }

```

Output

```

/tmp/Rxr66LcUHY.o
Enter the array size:6
Enter the array elements:0 1 2 3 4 5
1 3 5
*** Code Execution Successful ***

```

## 18.print the unique elements in an array

The screenshot shows a code editor interface with a dark theme. The left pane contains the C code for problem 18. The right pane shows the output of the compiled program.

```
main.c
1 #include<stdio.h>
2 #define MAX_SIZE 1000
3 int main()
4 {
5     int i,n,arr[MAX_SIZE],count,j;
6     printf("Enter the size:");
7     scanf("%d",&n);
8     printf("Enter the elements of the array.");
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    for(i=0;i<n;i++)
14    {
15        count=1;
16        for(j=0;j<n;j++)
17        {
18            if(arr[i]==arr[j])
19            {
20                count++;
21            }
22        }
23
24        if(count==2)
25        {
26            printf("%d ",arr[i]);
27        }
28    }
29    return 0;
30 }
```

Output:

```
/tmp/rZfV1icMba.o
Enter teh size:15
Enter the elements of the array:1 2 3 4 5 6 6 7 7 8 8 9 9 10 10
1 2 3 4 5
*** Code Execution Successful ***
```

## 19.print the second largest number in an array:

The screenshot shows a code editor interface with a dark theme. The left pane contains the C code for problem 19. The right pane shows the output of the compiled program.

```
main.c
1 //print the second largest element
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],max,second;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    max=arr[0];
15    second=0;
16    for(i=0;i<n;i++)
17    {
18        if(arr[i]>max)
19        {
20            second=max;
21            max=arr[i];
22        }
23        else if(arr[i]>second&&arr[i]<max)
24        {
25            second=arr[i];
26        }
27    }
28    printf("the second largest is:%d",second);
29    return 0;
30 }
```

Output:

```
/tmp/wIAVRXrqZH.o
Enter the size of the array:5
Enter the elements:1 2 4 4 5
the second largest is:4
*** Code Execution Successful ***
```

## 20.Print the average of the array

```

main.c | Run | Output
1 //average of an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],sum=0;
7     float avg;
8     printf("Enter the size of the array:");
9     scanf("%d",&n);
10    printf("Enter the elements:");
11    for(i=0;i<n;i++)
12    {
13        scanf("%d",&arr[i]);
14    }
15    for(i=0;i<n;i++)
16    {
17        sum+=arr[i];
18    }
19    avg=(float)sum/n;
20    printf("The average is:%f",avg);
21    return 0;
22 }

```

/tmp/D6z4i1mdJh.o  
Enter the size of the array:10  
Enter the elements:98 56 100 89 90 94 96 68 78 95  
The average is:86.40000  
==== Code Execution Successful ===

## 21.Count of array elements divisible by specific number

```

main.c | Run | Output
1 //Count of array elements divisible by specific number
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,k,arr[MAX_SIZE],count=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("Enter the divisible element:");
15    scanf("%d",&k);
16    for(i=0;i<n;i++)
17    {
18        if(arr[i]%k==0)
19        {
20            count++;
21        }
22    }
23    printf("the count:%d",count);
24    return 0;
25 }

```

/tmp/umm6Uzgbu1.o  
Enter the size of the array:5  
Enter the elements of the array:1 2 3 4 5  
Enter the divisible element:5  
the count:1  
==== Code Execution Successful ===

## 22.Frequency of a number

```

main.c | Run | Output
1 //count the frequency of the number
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,count,j,arr[MAX_SIZE];
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        count=1;
17        if(arr[i]!=-1)
18        {
19            for(j=i+1;j<n;j++)
20            {
21                if(arr[i]==arr[j])
22                {
23                    arr[j]=-1;
24                    {
25                        count++;
26                    }
27                }
28            }
29            printf("The frequency of %d is %d\n",arr[i],count);
30        }
31    }
32    return 0;
33 }

```

/tmp/Vzz8VkuaxE.o  
Enter the size of the array:5  
Enter the elements of the array:11 22 44 33 33  
The frequency of 11 is 1  
The frequency of 22 is 1  
The frequency of 44 is 1  
The frequency of 33 is 2  
==== Code Execution Successful ===

## 23.Sum of even numbers in an array

main.c				Output
<pre>1 //sum of even numbers in an array 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,n,arr[MAX_SIZE],sum=0; 7     printf("enter the size of the array:"); 8     scanf("%d",&amp;n); 9     printf("enter the elements of the array:"); 10    for(i=0;i&lt;n;i++) 11    { 12        scanf("%d",&amp;arr[i]); 13    } 14    for(i=0;i&lt;n;i++) 15    { 16        if(arr[i]%2==0) 17        { 18            sum=sum+arr[i]; 19        } 20    } 21    printf("The sum is:%d",sum); 22    return 0; 23 }</pre>	/tmp/lgbueR8wwk.o enter the size of the array:6 enter the elements of the array:2 4 6 2 2 2 The sum is:18 ==== Code Execution Successful ===			

## 24.To sort an array

main.c				Output
<pre>1 //to sort an array 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,j,n,arr[MAX_SIZE],temp; 7     printf("Enter the size of the array:"); 8     scanf("%d",&amp;n); 9     printf("Enter the elements of the array:"); 10    for(i=0;i&lt;n;i++) 11    { 12        scanf("%d",&amp;arr[i]); 13    } 14    for(i=0;i&lt;n;i++) 15    { 16        for(j=i+1;j&lt;n;j++) 17        { 18            if(arr[i]&gt;arr[j]) 19            { 20                temp=arr[i]; 21                arr[i]=arr[j]; 22                arr[j]=temp; 23            } 24        } 25    } 26    for(i=0;i&lt;n;i++) 27    { 28        printf("%d ",arr[i]); 29    } 30    return 0; 31 }</pre>	/tmp/TY24VNpvc1.o Enter the size of the array:6 Enter the elements of the array:5 9 4 48 62 84 4 5 9 48 62 84 ==== Code Execution Successful ===			

## 25.Sort the array in ascending order and print even numbers first and odd numbers next

main.c

```

1 //to sort an array and print even numbers first and odd numbers next
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,j,arr[MAX_SIZE],temp;
7     printf("Enter the size:");
8     scanf("%d",&n);
9     printf("Enter the elements:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        for(j=i+1;j<n;j++)
17        {
18            if(arr[i]>arr[j])
19            {
20                temp=arr[i];
21                arr[i]=arr[j];
22                arr[j]=temp;
23            }
24        }
25    }
26    printf("After sorting:");
27    for(i=0;i<n;i++)
28    {
29        printf("%d ",arr[i]);
30    }
31    printf("\nEven:");
32    for(i=0;i<n;i++)
33    {
34    }
35    if(arr[i]&1==0)
36    {
37        printf("%d ",arr[i]);
38    }
39    }
40    printf("\nOdd:");
41    for(i=0;i<n;i++)
42    {
43    }
44    if(arr[i]&1!=0)
45    {
46        printf("%d ",arr[i]);
47    }
48    }
49    return 0;
50 }
```

Output

```

/tmp/TDnaGMta.o
Enter the size:5
Enter the elements:1 9 6 5 2
after sorting:1 2 5 6 9
even:2 6
odd:1 5 9
== Code Execution Successful ==

```

## 26.Sum of duplicates in an array

main.c

```

1 //sum of duplicates in an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,j,arr[MAX_SIZE],sum=0;
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        if(arr[i]==-1)
17        {
18            for(j=i+1;j<n;j++)
19            {
20                if(arr[i]==arr[j])
21                {
22                    sum+=arr[i];
23                    arr[j]=-1;
24                }
25            }
26        }
27    }
28    printf("sum:%d",sum);
29    return 0;
30 }
```

Output

```

/tmp/qJ0s33DLm0.o
Enter the size of the array:7
Enter the elements of the array:1 4 2 5 2 4 5
sum:11
== Code Execution Successful ==

```

## 27.print the peak elements

```

main.c
1 //print the peak elements
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE];
7     printf("Enter the size of the array:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    printf("The peak elements are:");
15    for(i=0;i<n;i++)
16    {
17        if((i==0||arr[i]>arr[i-1])&&(i==n-1||arr[i]>arr[i+1]))
18        {
19            printf(" %d ",arr[i]);
20        }
21    }
22    return 0;
23 }

```

Output:

```

/tmp/gQjtDj2dlZ.o
Enter the size of the array:12
enter the elements of the array:1 5 7 96 48 25 3 57 74 2 6 4
the peak elements are:96 74 6
*** Code Execution Successful ***

```

## 28.Print the median of an array after sorting

```

main.c
1 //print the median
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],temp,j,sum=0;
7     printf("Enter the size:");
8     scanf("%d",&n);
9     printf("Enter the elements of the array:");
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13    }
14    for(i=0;i<n;i++)
15    {
16        for(j=i+1;j<n;j++)
17        {
18            if(arr[i]>arr[j])
19            {
20                temp=arr[i];
21                arr[i]=arr[j];
22                arr[j]=temp;
23            }
24        }
25    }
26    printf("After sorting:");
27    for(i=0;i<n;i++)
28    {
29        printf(" %d ",arr[i]);
30    }
31    printf("\n");
32    if(n%2==0)
33    {
34        sum=(arr[(n/2)]+arr[(n/2)-1])/2;
35    }
36    else
37    {
38        sum=arr[(n/2)];
39    }
40    printf("sum:%d ",sum);
41
42    return 0;
43 }

```

Output:

```

/tmp/g2aZK0G5x.o
Enter the size:5
Enter the elements of the array:1 5 9 3 4
After sorting:1 3 4 5 9
sum:4
*** Code Execution Successful ***

```

## 29.print middle elements in an array.

main.c

```

1 #include<stdio.h>
2 #define MAX_SIZE 1000
3 int main() {
4
5     int size,i,mid,arr[MAX_SIZE],mid1,mid2;
6     printf("Enter the size of the array: ");
7     scanf("%d", &size);
8     printf("Enter the elements of the array: ");
9     for (i = 0; i < size; i++) {
10         scanf("%d", &arr[i]);
11     }
12     if(size%2!=0)
13     {
14         mid = size / 2;
15         printf("The middle element is %d\n", arr[mid]);
16     }
17     else
18     {
19         mid1 = arr[size / 2 - 1];
20         mid2 = arr[size / 2];
21
22         int result[2] = {mid1, mid2};
23
24         printf("The two middle elements are: %d and %d\n", mid1, mid2);
25     }
26     return 0;
27 }
```

Output

```

/tmp/VM3azYL9Pt.o
Enter the size of the array: 5
Enter the elements of the array: 1 2 3 4 5
The middle element is 3

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

### 30.Median of an array with float and accurate median answer:

main.c

```

2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,j,arr[MAX_SIZE],temp,med2;
7     float med1;
8     scanf("%d",&n);
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    for(i=0;i<n;i++)
14    {
15        for(j=i+1;j<n;j++)
16        {
17            if(arr[i]>arr[j])
18            {
19                temp=arr[i];
20                arr[i]=arr[j];
21                arr[j]=temp;
22            }
23        }
24    }
25    for(i=0;i<n;i++)
26    {
27        printf("%d ",arr[i]);
28    }
29    if(n%2==0)
30    {
31        med1=(float)(arr[n/2]+arr[(n/2)-1])/2.0;
32        printf("\n%.2f",med1);
33    }
34    else
35    {
36        med2=arr[n/2];
37        printf("\n%d",med2);
38    }
39    return 0;
40 }
```

Output

```

/tmp/atQxgiUDex.o
4
23 45 67 89
23 45 67 89
56.00

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

### 31.removes all positive elements .If no negative elements are present, it prints "No negative elements present."

main.c

```
1 //display negative elements
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],neg=0;
7     scanf("%d",&n);
8     for(i=0;i<n;i++)
9     {
10         scanf("%d",&arr[i]);
11     }
12
13     for(i=0;i<n;i++)
14     {
15         if(arr[i]<0)
16         {
17             printf("%d ",arr[i]);
18             neg=1;
19         }
20     }
21     if(neg==0)
22     {
23         printf( "No negative elements present.");
24     }
25     return 0;
26 }
```

Output

```
/tmp/mBQYjxsxoZ.o
5
1 2 3 4 5
No negative elements present.

== Code Execution Successful ==
```

### 32.sum of first and last second first and second last

main.c

```
1 //sum of first and last second first and second last
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],sum=0;
7     int sum1=0,sum2=0;
8     scanf("%d",&n);
9     if(n<4)
10    {
11        printf("Invalid");
12    }
13    else
14    {
15        for(i=0;i<n;i++)
16        {
17            scanf("%d",&arr[i]);
18        }
19        sum1=arr[0]+arr[n-1];
20        sum2=arr[1]+arr[n-2];
21        sum=sum1+sum2;
22        printf("\n%d",sum1);
23        printf("\n%d",sum2);
24        printf("\n%d",sum);
25    }
26    return 0;
27 }
```

Output

```
/tmp/BSSqQ5Ljh0.o
2
Invalid

== Code Execution Successful ==
```

### 33.specific number divisor

The screenshot shows a code editor interface with a dark theme. On the left is a file browser sidebar with icons for various file types. The main area contains a C code file named 'main.c'. The code reads two integers from the user, scans for common divisors, and prints the count or a message if none exist. The output window shows the execution was successful.

```
main.c
1 //specific number divisor
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],count=0,k;
7     scanf("%d",&n);
8     for(i=0;i<n;i++)
9     {
10         scanf("%d",&arr[i]);
11     }
12     scanf("%d",&k);
13     for(i=0;i<n;i++)
14     {
15         if(arr[i]%k==0)
16         {
17             count++;
18         }
19     }
20     printf("%d",count);
21     if(count==n)
22     {
23         printf("\nCommon divisor available");
24     }
25     else
26     {
27         printf("\nNo Common divisor");
28     }
29     return 0;
30 }
```

Output

```
/tmp/pDmeumMnVs.o
5
49 21 28 35 14
2
2
No Common divisor
--- Code Execution Successful ---
```

### 34.calculate sum and product of all the elements in the array

The screenshot shows a code editor interface with a dark theme. The file browser sidebar is visible on the left. The main area contains a C code file named 'main.c'. The program reads the size of the array and its elements, then calculates and prints the sum and product. The output window shows the execution was successful.

```
main.c
1 //calculate sum and product of all the elements in the array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE];
7     int sum=0,prod=1;
8     scanf("%d",&n);
9     if(n<2)
10    {
11        printf("Invalid");
12    }
13    else
14    {
15        for(i=0;i<n;i++)
16        {
17            scanf("%d",&arr[i]);
18            sum=sum+arr[i];
19            prod=prod*arr[i];
20        }
21        printf("%d",sum);
22        printf("\n%d",prod);
23    }
24    return 0;
25 }
```

Output

```
/tmp/AKccHAAw0o.o
5
1 6 7 8 9
31
3024
--- Code Execution Successful ---
```

### 35.sum to the power of array size

The screenshot shows a dark-themed online C compiler interface. On the left, there's a vertical toolbar with icons for various languages and file operations. The main area has tabs for "main.c" and other files like "main.cpp", "main.java", etc. The "main.c" tab is active. The code in "main.c" is as follows:

```

1 //sum to the power of array size
2 #include<stdio.h>
3 #include<math.h>
4 #define MAX_SIZE 1000
5 int main()
6 {
7     int i,n,arr[MAX_SIZE],sum=0;
8     double power;
9     scanf("%d",&n);
10    for(i=0;i<n;i++)
11    {
12        scanf("%d",&arr[i]);
13        sum=sum+arr[i];
14    }
15    power=pow(sum,n);
16    printf("%d",sum);
17    printf("\n%.2lf",power);
18    return 0;
19 }

```

At the top right, there are buttons for "Run" and "Output". The "Output" section shows the following results:

```

/tmp/G7cjPuShmN.o
5
1 2 3 4 5
15
759375.00
== Code Execution Successful ==

```

### 36.sum of array and sum of array digits - division

This screenshot shows the same dark-themed online C compiler interface as the previous one. The "main.c" tab is active, displaying the following code:

```

1 //sum of array and sum of array digits - division
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE],sum1=0,sum2=0,num;
7     float res;
8     scanf("%d",&n);
9     for(i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    for(i=0;i<n;i++)
14    {
15        sum1=sum1+arr[i];
16    }
17    num=sum1;
18    while(num!=0)
19    {
20        sum2=sum2+num%10;
21        num=num/10;
22    }
23    res=(float)sum1/sum2;
24    printf("%.2f",res);
25    return 0;
26 }

```

The "Output" section shows the following results:

```

/tmp/SifqC6LBFt.o
5
1 2 3 4 5
2.50
== Code Execution Successful ==

```

### 37.adding k to each element

<b>main.c</b> <pre> 1 //adding k to each element 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,n,arr[MAX_SIZE],k; 7     scanf("%d",&amp;n); 8     for(i=0;i&lt;n;i++) 9     { 10         scanf("%d",&amp;arr[i]); 11     } 12     scanf("%d",&amp;k); 13     for(i=0;i&lt;n;i++) 14     { 15         arr[i]=arr[i]+k; 16         printf("%d ",arr[i]); 17     } 18     return 0; 19 }</pre>	<span style="border: 1px solid #ccc; padding: 2px;">[ ]</span> <span style="border: 1px solid #ccc; padding: 2px;">[ ]</span> <span style="border: 1px solid #ccc; padding: 2px;">Share</span> <span style="background-color: #0072bc; color: white; border: 1px solid #0072bc; padding: 2px 10px; border-radius: 5px;">Run</span>	<b>Output</b> <pre>/tmp/eeSECNLr75.o 5 1 9 7 5 3 1 2 10 8 6 4 == Code Execution Successful ==</pre>
---	--	--

### 38.identify the index of highest and lowest score

<b>main.c</b> <pre> 1 //identify the index of highest and lowest score 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,n,arr[MAX_SIZE],max,min,maxin=0,minin=0; 7     scanf("%d",&amp;n); 8     for(i=0;i&lt;n;i++) 9     { 10         scanf("%d",&amp;arr[i]); 11     } 12     min=arr[0]; 13     max=arr[0]; 14     for(i=0;i&lt;n;i++) 15     { 16         if(arr[i]&gt;max) 17         { 18             max=arr[i]; 19             maxin=i; 20         } 21         if(arr[i]&lt;min) 22         { 23             min=arr[i]; 24             minin=i; 25         } 26     } 27     printf("%d",maxin); 28     printf("\n%d",minin); 29     return 0; 30 }</pre>	<span style="border: 1px solid #ccc; padding: 2px;">[ ]</span> <span style="border: 1px solid #ccc; padding: 2px;">[ ]</span> <span style="border: 1px solid #ccc; padding: 2px;">Share</span> <span style="background-color: #0072bc; color: white; border: 1px solid #0072bc; padding: 2px 10px; border-radius: 5px;">Run</span>	<b>Output</b> <pre>/tmp/ukhY9vV28N.o 6 12 58 96 42 4 110 5 4 == Code Execution Successful ==</pre>
---	--	---

### 39.rounding mechanism

<pre>main.c</pre> <pre> 1 //rounding mechanism 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,n; 7     double arr[MAX_SIZE]; 8     scanf("%d",&amp;n); 9     for(i=0;i&lt;n;i++) 10    { 11        scanf("%lf",&amp;arr[i]); 12    } 13    for(i=0;i&lt;n;i++) 14    { 15        if(arr[i]==(int)arr[i]) 16        { 17            printf("%.0lf ",arr[i]); 18        } 19    } 20    return 0; 21 }</pre>	   	<p><b>Output</b></p> <pre>/tmp/AxGP200R4Z.o 5 99.5 95.4 100.0 99 54.5 100 99 ==== Code Execution Successful ===</pre>
--	--	---

#### 40.print unique elements along with their sum

<pre>main.c</pre> <pre> 1 //print the unique elements 2 #include&lt;stdio.h&gt; 3 #define MAX_SIZE 1000 4 int main() 5 { 6     int i,n,arr[MAX_SIZE],sum=0,count,j; 7     scanf("%d",&amp;n); 8     for(i=0;i&lt;n;i++) 9     { 10        scanf("%d",&amp;arr[i]); 11    } 12    for(i=0;i&lt;n;i++) 13    { 14        count=1; 15        for(j=0;j&lt;n;j++) 16        { 17            if(arr[i]==arr[j]) 18            { 19                count++; 20            } 21        } 22        if(count==2) 23        { 24            printf("%d ",arr[i]); 25            sum=sum+arr[i]; 26        } 27    } 28    printf("\n%d",sum); 29    return 0; 30 }</pre>	   	<p><b>Output</b></p> <pre>/tmp/06btb5YuhB.o 9 9 8 7 6 9 8 7 10 11 6 10 11 27 ==== Code Execution Successful ===</pre>
---	---	---

#### 41.Increment each element by 1

```

main.c
1 //increment 1 to each elemnt in an array
2 #include<stdio.h>
3 #define MAX_SIZE 1000
4 int main()
5 {
6     int i,n,arr[MAX_SIZE];
7     scanf("%d",&n);
8     for(i=0;i<n;i++)
9     {
10         scanf("%d",&arr[i]);
11     }
12     for(i=0;i<n;i++)
13     {
14         arr[i]=arr[i]+1;
15         printf("%d ",arr[i]);
16     }
17     return 0;
18 }
```

Output

```

/tmp/VfNADdb6Lp.o
7
15 24 98 75 41 85 62
16 25 99 76 42 86 63
--- Code Execution Successful ---
```

## MATRIX

### 42.program to add two matrices

```

1 //C program to add two matrices
2 #include<stdio.h>
3 #define n 3
4 int main()
5 {
6     int A[n][n];
7     int B[n][n];
8     int C[n][n];
9     int i,j;
10    for(i=0;i<n;i++)
11    {
12        for(j=0;j<n;j++)
13        {
14            scanf("%d",&A[i][j]);
15        }
16    }
17    for(i=0;i<n;i++)
18    {
19        for(j=0;j<n;j++)
20        {
21            scanf("%d",&B[i][j]);
22        }
23    }
24    for(i=0;i<n;i++)
25    {
26        for(j=0;j<n;j++)
27        {
28            C[i][j]=A[i][j]+B[i][j];
29        }
30    }
31    for(i=0;i<n;i++)
32    {
33        for(j=0;j<n;j++)
34        {
35            printf("%d ",C[i][j]);
36        }
37        printf("\n");
38    }
39    return 0;
40 }
```

Output

```

/tmp/b89M6e0uuH.o
1 2 3
4 5 6
7 8 9
8 9 4
7 6 1
2 3 4
9 11 7
11 11 7
9 11 13
--- Code Execution Successful ---
```

### 43.Scalar Matrix Multiplication

main.c

1 //Program to perform scalar matrix multiplication  
2 #include<stdio.h>  
3 #define n 3  
4 int main()  
5 {  
6 int A[n][n];  
7 int num,i,j;  
8 for(i=0;i<n;i++)  
9 {  
10 for(j=0;j<n;j++)  
11 {  
12 scanf("%d",&A[i][j]);  
13 }  
14 }  
15 scanf("%d",&num);  
16 for(i=0;i<n;i++)  
17 {  
18 for(j=0;j<n;j++)  
19 {  
20 A[i][j]=num\*A[i][j];  
21 printf("%d ",A[i][j]);  
22 }  
23 printf("\n");  
24 }  
25  
26 return 0;  
27 }

/tmp/IpbvfFgtp  
1 2 3  
4 5 6  
7 8 9  
3  
3 6 9  
12 15 18  
21 24 27  
==== Code Executed

**44.to check whether the matrix is symmetric or not**

main.c

```
1 // C PROGRAM TO CHECK WHETHER THE GIVEN MATRIX IS SYMMETRIC OR NOT
2 #include<stdio.h>
3 #define n 3
4 int main()
5 {
6     int i,j,issymmetric;
7     int A[n][n];
8     int B[n][n];
9     for(i=0;i<n;i++)
10    {
11        for(j=0;j<n;j++)
12        {
13            scanf("%d", &A[i][j]);
14        }
15    }
16    for(i=0;i<n;i++)
17    {
18        for(j=0;j<n;j++)
19        {
20            B[i][j]=A[j][i];
21        }
22    }
23    issymmetric=1;
24    for(i=0;i<n&&issymmetric;i++)
25    {
26        for(j=0;j<n;j++)
27        {
28            if(A[i][j]!=B[i][j])
29            {
30                issymmetric=0;
31                break;
32            }
33        }
34    }
35    if(issymmetric==1)
36    {
37        printf("Symmetric\n");
38        for(i=0;i<n;i++)
39        {
40            for(j=0;j<n;j++)
41            {
42                printf("%d ",A[i][j]);
43            }
44            printf("\n");
45        }
46    }
47    else
48    {
49        printf("Not symmetric");
50    }
51    return 0;
52 }
```

Output

```
/tmp/qKagh7fg05.o
1 2 3
4 5 6
7 8 9
Not symmetric
*** Code Execution Successful ***
```

```

main.c
1 //to verify whether the matrix is symmetric or not
2 #include<stdio.h>
3 int main()
4 {
5     int i,j,n,issymmetric;
6     scanf("%d",&n);
7     int A[n][n];
8     int B[n][n];
9     for(i=0;i<n;i++)
10    {
11        for(j=0;j<n;j++)
12        {
13            scanf("%d",&A[i][j]);
14        }
15    }
16    for(i=0;i<n;i++)
17    {
18        for(j=0;j<n;j++)
19        {
20            B[i][j]=A[j][i];
21        }
22    }
23    issymmetric=1;
24    for(i=0;i<n;issymmetric;i++)
25    {
26        for(j=0;j<n;j++)
27        {
28            if(A[i][j]==B[i][j])
29            {
30                issymmetric=0;
31                break;
32            }
33        }
34    }
35    if(issymmetric==1)
36    {
37        printf("symmetric\n");
38        for(i=0;i<n;i++)
39        {
40            for(j=0;j<n;j++)
41            {
42                printf("%d ",A[i][j]);
43            }
44            printf("\n");
45        }
46    }
47 }
48 else
49 {
50     printf("not symmetric");
51 }
52 return 0;
53 }

```

Output  
`/tmp/AuhdrIkSxP.o  
2  
1 2  
2 1  
symmetric  
1 2  
2 1  
*** Code Execution Successful ***`

## 45. To check sparse matrix or not

```

main.c
1 //to check whether the matrix is sparse or not
2 #include<stdio.h>
3 int main()
4 {
5     int i,j,n,total=0;
6     scanf("%d",&n);
7     int A[n][n];
8     for(i=0;i<n;i++)
9     {
10        for(j=0;j<n;j++)
11        {
12            scanf("%d",&A[i][j]);
13        }
14    }
15    for(i=0;i<n;i++)
16    {
17        for(j=0;j<n;j++)
18        {
19            if(A[i][j]==0)
20            {
21                total++;
22            }
23        }
24    }
25    if(total>i*j/2)
26    {
27        printf("Sparse");
28    }
29    else
30    {
31        printf("not sparse");
32    }
33 return 0;
34 }

```

Output  
`/tmp/Y5eZMTtLf3.o  
4  
1 1 1  
1 1 0 0  
0 0 0 0  
0 0 0 0  
Sparse  
*** Code Execution Successful ***`

## 46.To find the determinant of the matrix

The screenshot shows a code editor interface with a dark theme. On the left is the code file 'main.c' containing C code to calculate the determinant of a 3x3 matrix. On the right are buttons for copy, share, and run, and an 'Output' window showing the results of the execution.

```
1 #include<stdio.h>
2 #define n 3
3 int main()
4 {
5     int i,j,a,b,c,d,e,f,g,h,k;
6     int A[n][n];
7     long int det;
8     for(i=0;i<n;i++)
9     {
10         for(j=0;j<n;j++)
11         {
12             if(scanf("%d",&A[i][j])!=1)
13             {
14                 printf("Invalid input\n");
15                 return 1;
16             }
17         }
18     }
19     a=A[0][0];
20     b=A[0][1];
21     c=A[0][2];
22     d=A[1][0];
23     e=A[1][1];
24     f=A[1][2];
25     g=A[2][0];
26     h=A[2][1];
27     k=A[2][2];
28     det=(a*(e*k-f*h))-(b*(d*k-f*g))+(c*(d*h-e*g));
29     for(i=0;i<n;i++)
30     {
31         for(j=0;j<n;j++)
32         {
33             printf("%d ",A[i][j]);
34         }
35         printf("\n");
36     }
37     printf("\n%d",det);
38     return 0;
39 }
```

Output:

```
/tmp/5cNJ7KTemP.o
6 1 1
4 -2 5
2 8 7
6 1 1
4 -2 5
2 8 7
-306
== Code Execution Successful ==
```

## 47.transpose of the given rows and columns

The screenshot shows a code editor interface with a dark theme. On the left is the code file 'main.c' containing C code to transpose a matrix. On the right are buttons for copy, share, and run, and an 'Output' window showing the results of the execution.

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,j;
5     int rows,cols;
6     scanf("%d",&rows);
7     scanf("%d",&cols);
8     if(rows<0||cols<0||rows>20||cols>20)
9     {
10         printf("invalid input");
11         return 1;
12     }
13     int A[rows][cols];
14     int B[cols][rows];
15     for(i=0;i<rows;i++)
16     {
17         for(j=0;j<cols;j++)
18         {
19             scanf("%d",&A[i][j]);
20         }
21     }
22     for(i=0;i<rows;i++)
23     {
24         for(j=0;j<cols;j++)
25         {
26             B[j][i]=A[i][j];
27         }
28     }
29     for(i=0;i<rows;i++)
30     {
31         for(j=0;j<cols;j++)
32         {
33             printf("%d ",B[i][j]);
34         }
35         printf("\n");
36     }
37     for(i=0;i<cols;i++)
38     {
39         for(j=0;j<rows;j++)
40         {
41             printf("%d ",B[i][j]);
42         }
43         printf("\n");
44     }
45     return 0;
46 }
```

Output:

```
/tmp/T2054Hirj6V.o
4
2
1 2
3 4
5 6
7 8
1 2
3 4
5 6
7 8
1 3 5 7
2 4 6 8
== Code Execution Successful ==
```