✅ Day : Arrays (6-8-2025)

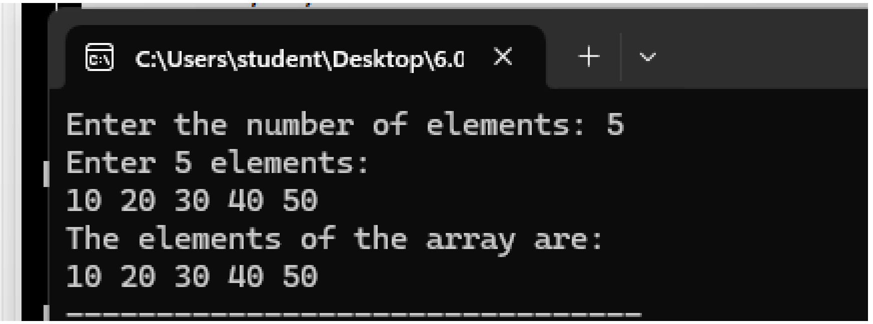
1. Write a program to read and print elements of an array.

- □ **Input: Size of the array n, and n elements.**

- □ **Process: Store each element into the array and then print them using a loop.**

- □ **Output: Elements of the array.**

```
#include <stdio.h>
int main()
{
   int n, i;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
{
      scanf("%d", &arr[i]);
   }
   printf("The elements of the array are:\n");
   for(i = 0; i < n; i++)
{
      printf("%d ", arr[i]);
   }
   return 0;
}
```



```
Enter the number of elements: 5
Enter 5 elements:
10 20 30 40 50
The elements of the array are:
10 20 30 40 50
--------------------------------
```

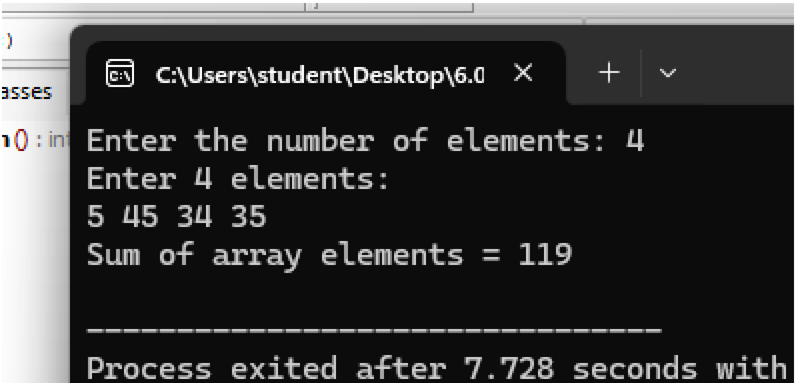
2. Write a program to find the sum of elements of an array.

□ **Input: Size of the array n, and n array elements.**

□ **Process: Add all the elements using a loop.**

□ **Output: Sum of all array elements**

```
#include <stdio.h>
int main()
{
   int n, i, sum = 0;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
{
      scanf("%d", &arr[i]);
      sum += arr[i];
   }
   printf("Sum of array elements = %d\n", sum);
   return 0;
}
```



```
Enter the number of elements: 4
Enter 4 elements:
5 45 34 35
Sum of array elements = 119

--------------------------------
Process exited after 7.728 seconds with
```

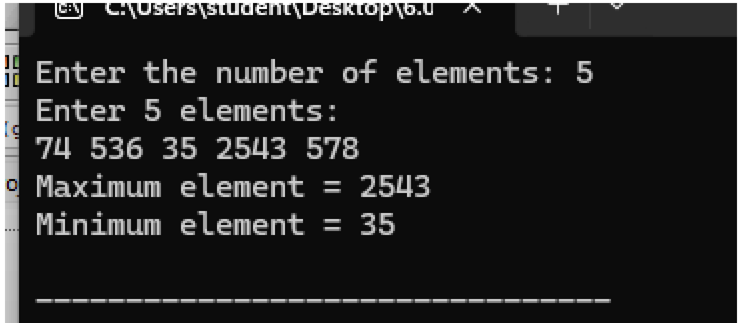
3. Write a program to find the maximum and minimum element in an array.

□ **Input: Size of the array n, and n array elements.**

□ **Process: Traverse the array to compare and update max and min.**

□ **Output: Maximum and minimum elements in the array.**

```
#include <stdio.h>
int main()
{
  int n, i;
  printf("Enter the number of elements: ");
  scanf("%d", &n);
  int arr[n];
  printf("Enter %d elements:\n", n);
  for(i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
  }
  int max = arr[0];
  int min = arr[0];
  for(i = 1; i < n; i++)
{
    if(arr[i] > max)
      max = arr[i];
    if(arr[i] < min)
      min = arr[i];
  }
  printf("Maximum element = %d\n", max);
  printf("Minimum element = %d\n", min);
  return 0;
}
```



```
C:\Users\student\Desktop\6.u

Enter the number of elements: 5
Enter 5 elements:
74 536 35 2543 578
Maximum element = 2543
Minimum element = 35

--------------------------------
```

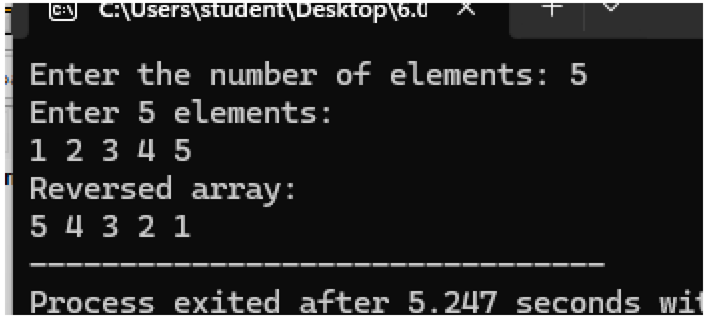
4. Write a program to reverse an array.

□ **Input: Size of the array n, and n array elements.**

□ **Process: Swap elements from both ends of the array.**

□ **Output: Array elements in reverse order.**

```
#include <stdio.h>
int main()
{
   int n, i;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
{
      scanf("%d", &arr[i]);
   }
   for(i = 0; i < n / 2; i++)
{
      int temp = arr[i];
      arr[i] = arr[n - 1 - i];
      arr[n - 1 - i] = temp;
   }
   printf("Reversed array:\n");
   for(i = 0; i < n; i++)
{
      printf("%d ", arr[i]);
   }
   return 0;
}
```



```
Enter the number of elements: 5
Enter 5 elements:
1 2 3 4 5
Reversed array:
5 4 3 2 1
--------------------------------
Process exited after 5.247 seconds wi
```

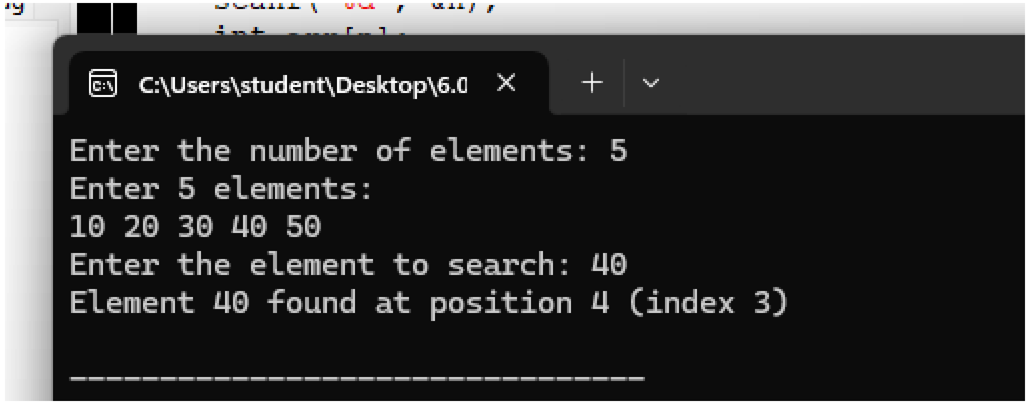
□ **5. Write a program to search for an element in an array (linear search).**

□ **Input: Size of array n, n array elements, and the element to search (key).**

□ **Process: Compare key with each element of the array sequentially.**

□ **Output: Position of the element if found, otherwise a message saying not found.**

```
#include <stdio.h>
int main()
{
  int n, i, key, found = 0;
  printf("Enter the number of elements: ");
  scanf("%d", &n);
  int arr[n];
  printf("Enter %d elements:\n", n);
  for(i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
  }
  printf("Enter the element to search: ");
  scanf("%d", &key);
  for(i = 0; i < n; i++)
{
    if(arr[i] == key)
{
      printf("Element %d found at position %d (index %d)\n", key, i + 1, i);
      found = 1;
      break;
    }
  }
  if(!found)
{
    printf("Element %d not found in the array.\n", key);
  }
  return 0;
}
```

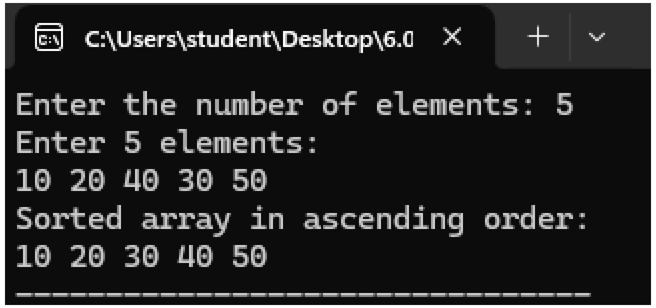


```
Enter the number of elements: 5
Enter 5 elements:
10 20 30 40 50
Enter the element to search: 40
Element 40 found at position 4 (index 3)

--------------------------------
```

6. Write a program to sort an array in ascending order.

- □ **Input: Size of the array n, and n array elements.**
- □ **Process: Compare and swap elements to sort in ascending order using Bubble Sort.**
- □ **Output: Sorted array elements.**

```
#include <stdio.h>
int main() {
  int n, i, j, temp;
  printf("Enter the number of
elements: ");
  scanf("%d", &n);

  int arr[n];

  printf("Enter %d elements:\n", n);
  for(i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
  }
  for(i = 0; i < n - 1; i++) {
    for(j = 0; j < n - 1 - i; j++)
{
      if(arr[j] > arr[j + 1])
{
        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
  printf("Sorted array in ascending
order:\n");
  for(i = 0; i < n; i++) {
    printf("%d ", arr[i]);
  }
  return 0;
}
```



```
Enter the number of elements: 5
Enter 5 elements:
10 20 40 30 50
Sorted array in ascending order:
10 20 30 40 50
--------------------------------
```

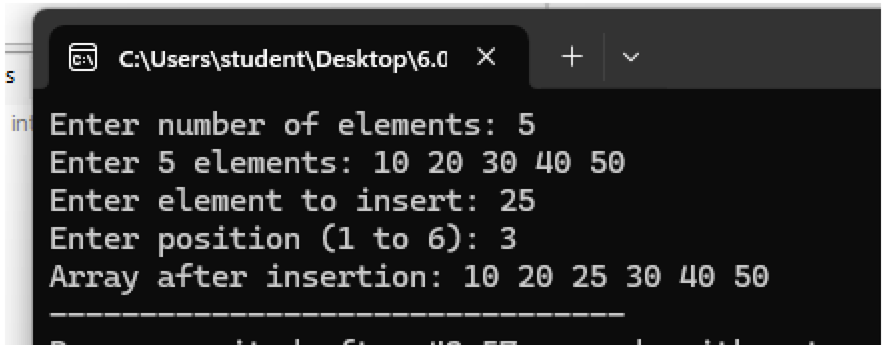
```
#include <stdio.h>
int main()
{
   int arr[100], n, i, pos, ele;
   printf("Enter number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements: ", n);
   for(i = 0; i < n; i++) scanf("%d", &arr[i]);
   printf("Enter element to insert: ");
   scanf("%d", &ele);
   printf("Enter position (1 to %d): ", n + 1);
   scanf("%d", &pos);
   if(pos < 1 || pos > n + 1)
{
      printf("Invalid position!");
      return 1;
   }
   for(i = n; i >= pos; i--)
      arr[i] = arr[i - 1];
   arr[pos - 1] = ele;
   n++;
   printf("Array after insertion: ");
   for(i = 0; i < n; i++) printf("%d ", arr[i]);

   return 0;
}
```

7. Write a program to insert an element in an array.

□ **Input: Array size n, array elements, element to insert `element`, and position pos.**

□ **Process: Shift elements to the right from the given position and insert the new element.**

□ **Output: Array after insertion**



```
Enter number of elements: 5
Enter 5 elements: 10 20 30 40 50
Enter element to insert: 25
Enter position (1 to 6): 3
Array after insertion: 10 20 25 30 40 50
--------------------------------
```

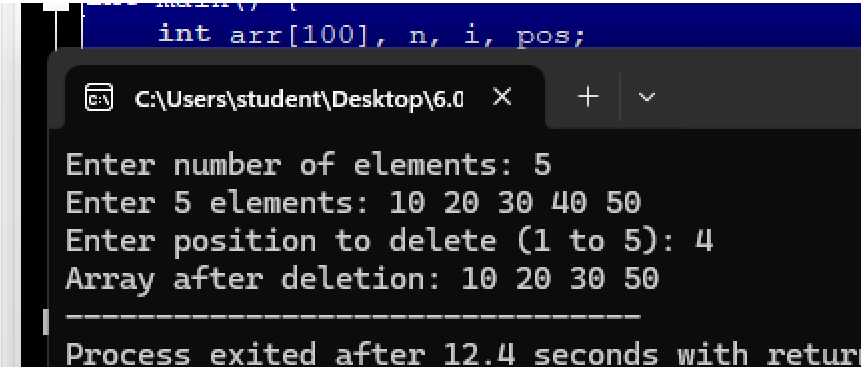
8. Write a program to delete an element from an array.

□ **Input: Array size n, n elements, and position pos to delete.**

□ **Process: Shift elements left from pos.**

□ **Output: Array after deletion.**

```
#include <stdio.h>
int main() {
  int arr[100], n, i, pos;
  printf("Enter number of elements: ");
  scanf("%d", &n);
  printf("Enter %d elements: ", n);
  for(i = 0; i < n; i++) scanf("%d", &arr[i]);
  printf("Enter position to delete (1 to %d): ", n);
  scanf("%d", &pos);
  if(pos < 1 || pos > n)
{
    printf("Invalid position!");
    return 1;
  }
  for(i = pos - 1; i < n - 1; i++)
    arr[i] = arr[i + 1];
  n--;
  printf("Array after deletion: ");
  for(i = 0; i < n; i++) printf("%d ", arr[i]);
  return 0;
}
```



```
    int arr[100], n, i, pos;
```

```
Enter number of elements: 5
Enter 5 elements: 10 20 30 40 50
Enter position to delete (1 to 5): 4
Array after deletion: 10 20 30 50
--------------------------------
Process exited after 12.4 seconds with retur
```

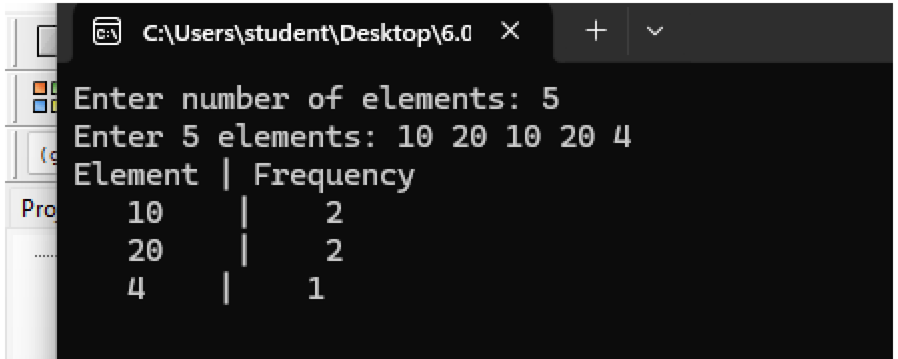
9. Write a program to find the frequency of elements in an array.

□ **Input: Array size n and n elements.**

□ **Process: Count how many times each unique element appears.**

□ **Output: Display each element with its frequency.**

```
#include <stdio.h>
int main()
{
  int arr[100], freq[100], n, i, j, count;
  printf("Enter number of elements: ");
  scanf("%d", &n);
  printf("Enter %d elements: ", n);
  for(i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
    freq[i] = -1;
  }
  for(i = 0; i < n; i++)
{
    if(freq[i] == -1)
{
      count = 1;
      for(j = i + 1; j < n; j++)
{
        if(arr[i] == arr[j])
{
          count++;
          freq[j] = 0;
        }
      }
      freq[i] = count;
    }
  }
  printf("Element | Frequency\n");
  for(i = 0; i < n; i++) {
    if(freq[i] != 0)
      printf("   %d   |   %d\n", arr[i],
freq[i]);
  }
  return 0;
```



```
Enter number of elements: 5
Enter 5 elements: 10 20 10 20 4
Element | Frequency
   10    |    2
   20    |    2
   4    |    1
```

□ **Write a program to merge two arrays.**

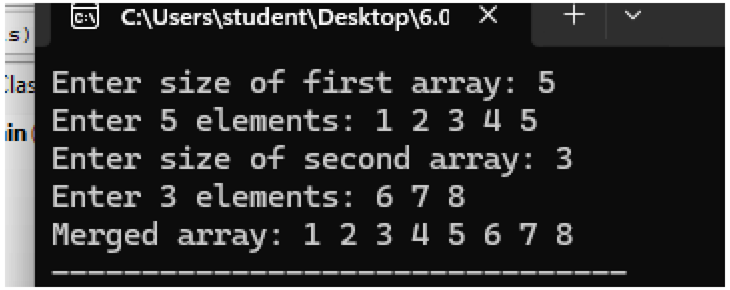
□ **Input: Sizes and elements of two arrays `arr1` and `arr2`**

□ **.**

□ **Process: Copy elements of both arrays into a third array.**

□ **Output: Merged array.**

```
#include <stdio.h>
int main()
{
   int arr1[50], arr2[50], merged[100];
   int n1, n2, i, k = 0;
   printf("Enter size of first array: ");
   scanf("%d", &n1);
   printf("Enter %d elements: ", n1);
   for(i = 0; i < n1; i++)
{
      scanf("%d", &arr1[i]);
      merged[k++] = arr1[i];
   }
   printf("Enter size of second array: ");
   scanf("%d", &n2);
   printf("Enter %d elements: ", n2);
   for(i = 0; i < n2; i++)
{
      scanf("%d", &arr2[i]);
      merged[k++] = arr2[i];
   }
   printf("Merged array: ");
   for(i = 0; i < n1 + n2; i++)
{
      printf("%d ", merged[i]);
   }
   return 0;
}
```



```
Enter size of first array: 5
Enter 5 elements: 1 2 3 4 5
Enter size of second array: 3
Enter 3 elements: 6 7 8
Merged array: 1 2 3 4 5 6 7 8
--------------------------------
```