Day 4: Arrays (6-8-2025)

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

IPO:

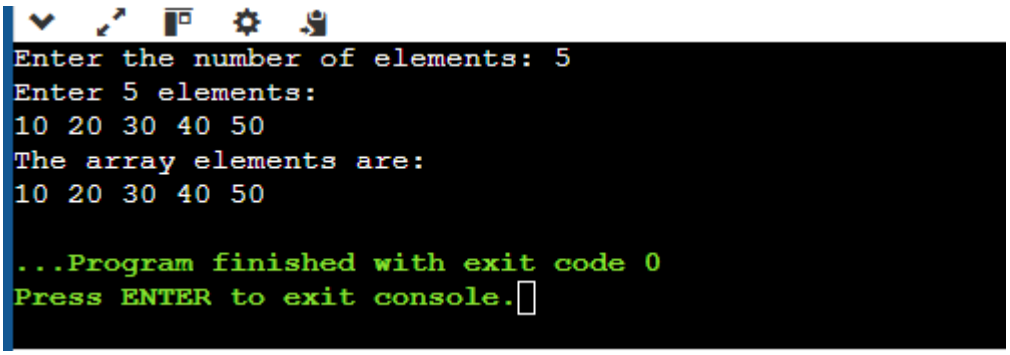
Input: taking the number of elements as n, and array elements.

Process: store input in an array then loop to display.

Output: display the elements of the array.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10],n,i;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements:\n", n);
   for (i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }
   printf("The array elements are:\n");
   for (i = 0; i < n; i++)
   {
      printf("%d ", arr[i]);
   }
}
```



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

IPO:

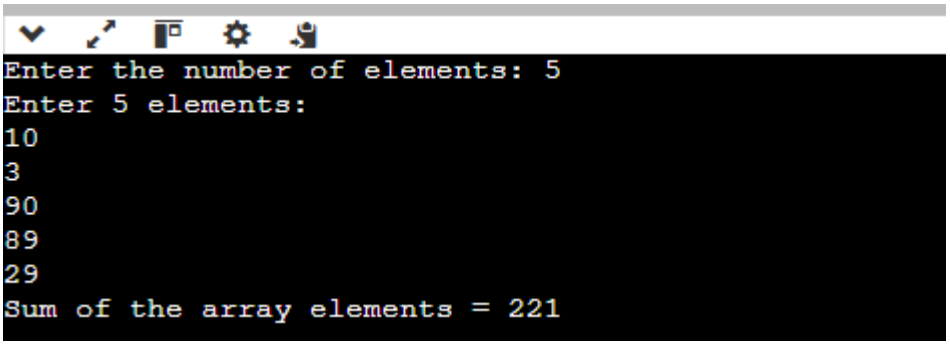
Input: taking the number of elements n and array elements.

Process: loop through the array to calculate the sum.

Output: print the sum of array elements.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10],n,i,s=0;
   printf("Enter the number of elements: ");
   scanf("%d",&n);
   printf("Enter %d elements:\n", n);
   for (i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
      s=s+arr[i];
   }
   printf("Sum of the array elements = %d\n",s);
}
```



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

IPO:

Input: numbers of elements n in an array elements.

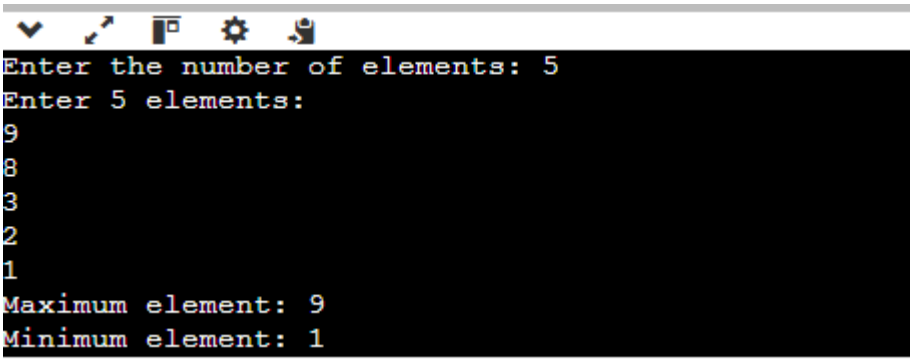
Process: loop through the array to calculate max,min.

Output: display the maximum and minimum.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10],n,i,max,min;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements:\n", n);
   for (i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }
   max = arr[0];
   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);
}
```



4. Write a program to reverse an array.

IPO:

Input: numbers of elements in array n and array elements.

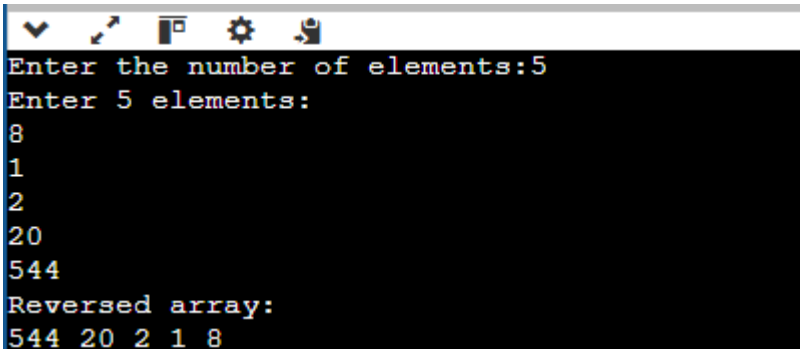
Process: loop through the array in reverse.

Output: print the array elements in reverse order.

Code:

```
#include <stdio.h>
void main()
{
    int arr[10],n,i;
```

```
    printf("Enter the number of elements:");
    scanf("%d",&n);
    printf("Enter %d elements:\n",n);
    for (i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Reversed array:\n");
    for (i=n-1; i>=0;i--)
    {
        printf("%d ", arr[i]);
    }
}
```



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

IPO:

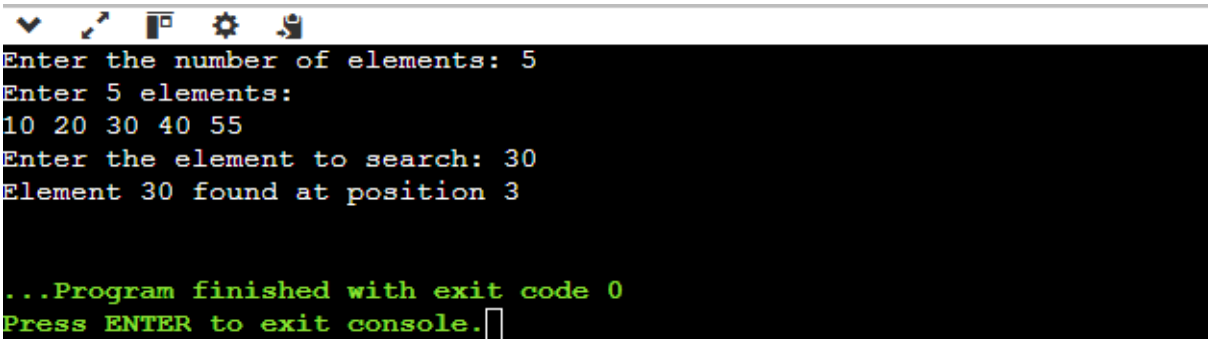
Input: numbers of elements n,and array elements ,with search value.

Process: compare each element with search value.

Output: position of elements are found or not found message.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10], n, i, search, found = 0;
   printf("Enter the number of elements: ");
   scanf("%d", &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", &search);
   for (i = 0; i < n; i++)
   {
      if (arr[i] == search)
      {
         found = 1;
         break;
      }
   }
   if (found)
      printf("Element %d found at position %d\n", search, i + 1);
   else
      printf("Element %d not found in the array.\n", search);
}
```



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

IPO:

Input: taking the numbers of elements n, and array of the elements.

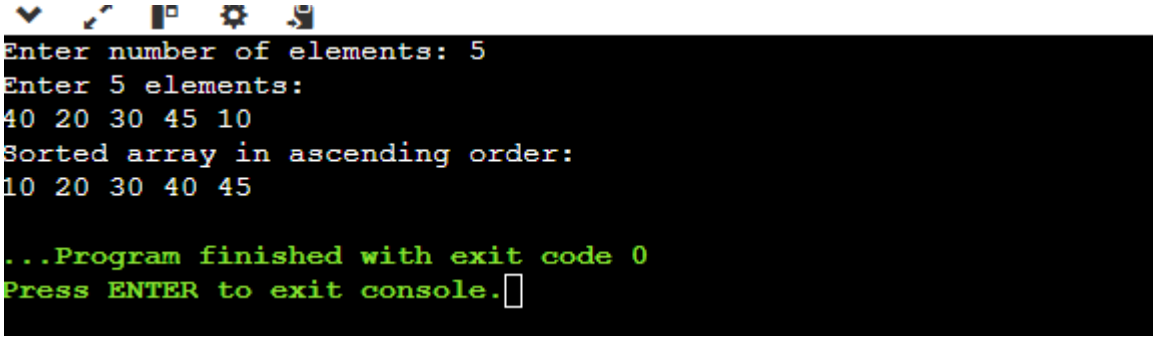
Process: Use bubble sort to compare and swap elements.

Output: array sorted in ascending order.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10], n, i, j, temp;
   printf("Enter number of elements: ");
   scanf("%d", &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-i-1; 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]);
    }
}
```



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

IPO:

Input: taking numbers of elements n, array elements position value.

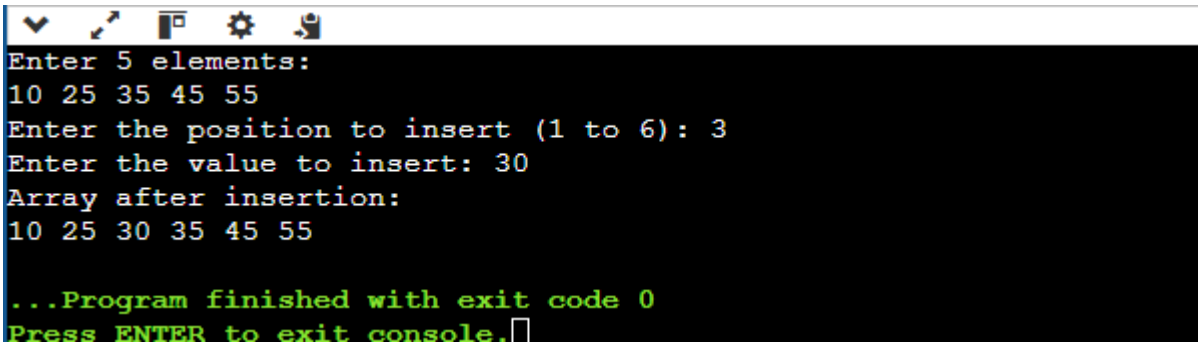
Process: shift elements from position, insert value.

Output: array with the new element inserted.

Code:

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

}



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

IPO:

Input: taking the elements from n, array elements, positions

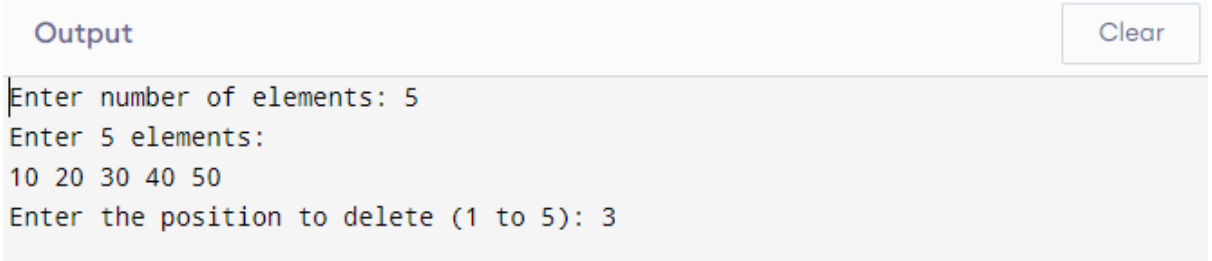
Process: remove the element by shifting elements by left.

Output: array after the deletion of the specified element.

Code:

```
#include <stdio.h>
void main()
{
   int arr[10], n, i, pos;
   printf("Enter number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }
   printf("Enter the position to delete (1 to %d): ", n);
   scanf("%d", &pos);
```

```
    if(pos < 1 || pos > n)
    {
        printf("Invalid position!\n");
    }
    else
    {
        for(i = pos - 1; i < n - 1; i++)
        {
            arr[i] = arr[i + 1];
        }
        n--;
        printf("Array after deletion:\n");
        for(i = 0; i < n; i++)
        {
            printf("%d ", arr[i]);
        }
    }
}
```



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

IPO:

Input: taking the array element.

Process: count how many times each number occurs.

Output: print frequency of each number.

Code:

```
#include <stdio.h>
void main()
{
   int a[10], n, i, j, count;
   printf("Enter size: ");
   scanf("%d", &n);
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
      scanf("%d", &a[i]);
      for(i = 0; i < n; i++)
      {
      count = 1;
      if(a[i] != -1)
      {
         for(j = i + 1; j < n; j++)
         {
            if(a[i] == a[j])
            {
               count++;
               a[j] = -1; // mark counted
            }
         }
         printf("%d occurs %d times\n", a[i], count);
      }
```

```
    }
}
```

10. Write a program to merge two arrays.

IPO:

Input: number of array a[5]= {1,2,3,4,5} and b[3]={6,7,8}.

Process: copy elements of both arrays into one array.

Output: merged array c[ ] ={1,2,3,4,5,6,7,8} .

Code:

```
#include <stdio.h>
void main()
{
    int a[5] = {1, 2, 3, 4, 5};
    int b[3] = {6, 7, 8};
    int c[8], i;
    for(i = 0; i < 5; i++)
        c[i] = a[i];
    for(i = 0; i < 3; i++)
        c[i + 5] = b[i];
    printf("Merged array:\n");
```

```
    for(i = 0; i < 8; i++)
        printf("%d ", c[i]);
}
```

Output Clear

```
Merged array:
1 2 3 4 5 6 7 8

=== Code Exited With Errors ===
```