Day 4 : Arrays (6-8-2025)

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

IPO

INPUT: Size of array (n), then n elements

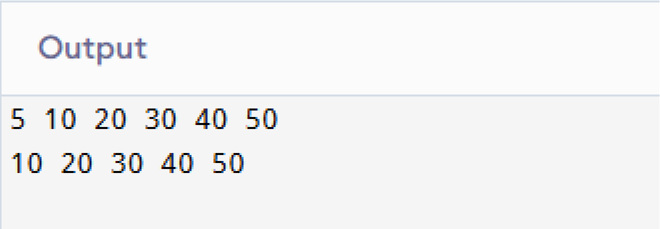
PROCESS: Store and print elements using loop

OUTPUT: Display all elements

CODE;

```
#include<stdio.h>
void main()
{
    int a[100], n, i;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
}
```

OUTPUT;



```
5 10 20 30 40 50
10 20 30 40 50
```

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

IPO

INPUT: Array size n and elements

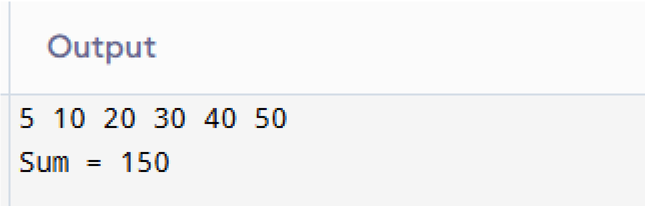
PROCESS: Use loop to calculate

OUTPUT: Sum of all elements

CODE;

```
#include<stdio.h>
void main()
{
    int a[50], n, i, sum = 0;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for(i = 0; i < n; i++)
        sum = sum+ a[i];
    printf("Sum = %d", sum);
}
```

OUTPUT;



```
5 10 20 30 40 50
Sum = 150
```

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

IPO

INPUT: n elements in array

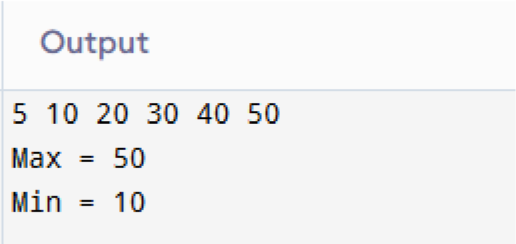
PROCESS: Compare each to track max & min

OUTPUT: Print maximum and minimum

CODE;

```
#include<stdio.h>
void main()
{
    int a[30], n, i, max, min;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);
    max = min = a[0];
    for(i = 1; i < n; i++)
    {
        if(a[i] > max)
            max = a[i];
        if(a[i] < min)
            min = a[i];
    }
    printf("Max = %d\n", max);
    printf("Min = %d", min);
}
```

OUTPUT;



```
5 10 20 30 40 50
Max = 50
Min = 10
```

4. Write a program to reverse an array.

IPO

INPUT: Array of n elements

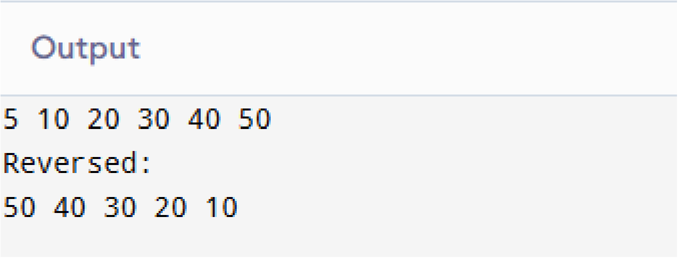
PROCESS: Print from end to start

OUTPUT: Reversed array

CODE;

```
#include<stdio.h>
void main()
{
    int a[20], n, i;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);
    printf("Reversed:\n");
    for(i = n - 1; i >= 0; i--)
        printf("%d ", a[i]);
}
```

OUTPUT:



```
5 10 20 30 40 50
Reversed:
50 40 30 20 10
```

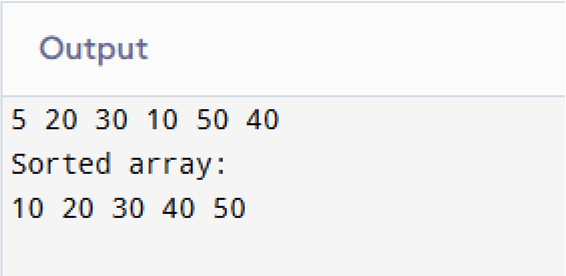
6. Write a program to sort an array in ascending order.
IPO

INPUT: Array size and elements
PROCESS: Use selection sort or bubble sort
OUTPUT: Sorted array

CODE:

```
#include<stdio.h>

void main()

{

    int a[100], n, i, j, temp;

    scanf("%d", &n);

    for(i = 0; i < n; i++)

        scanf("%d", &a[i]);

    for(i = 0; i < n-1; i++)

    {

        for(j = i+1; j < n; j++)

        {

            if(a[i] > a[j])

            {

                temp = a[i];

                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("Sorted array:\n");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
}
```

OUTPUT;



```
5 20 30 10 50 40
Sorted array:
10 20 30 40 50
```

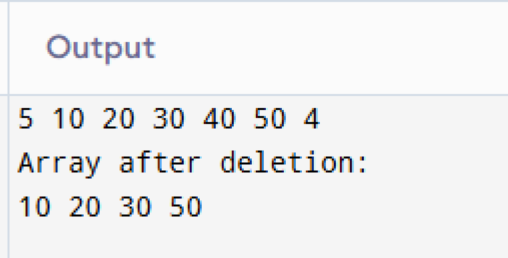
8.Write a program to delete an element from an array.

IPO
INPUT: Array and position to delete
PROCESS: Shift left from deleted position
OUTPUT: Updated array

CODE;

```
#include<stdio.h>

void main()

{

    int a[100], n, i, pos;

    scanf("%d", &n);

    for(i = 0; i < n; i++)

        scanf("%d", &a[i]);

    scanf("%d", &pos);

    for(i = pos - 1; i < n - 1; i++)

        a[i] = a[i + 1];

    n--;

    printf("Array after deletion:\n");

    for(i = 0; i < n; i++)

        printf("%d ", a[i]);

}
```

OUTPUT;



```
5 10 20 30 40 50 4
Array after deletion:
10 20 30 50
```

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

IPO

INPUT: Array of n elements

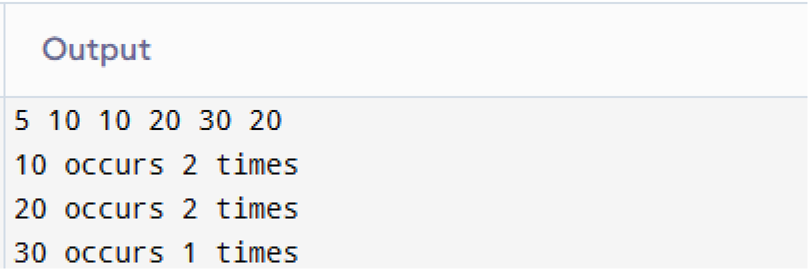
PROCESS: Count each unique number

OUTPUT: Frequency of each element

CODE;

```
#include<stdio.h>
void main()
{
    int a[100], freq[100], n, i, j, count;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
        freq[i] = -1;
    }
    for(i = 0; i < n; i++)
    {
        count = 1;
        for(j = i + 1; j < n; j++)
        {
            if(a[i] == a[j])
            {
                count++;
                freq[j] = 0;
            }
        }
        if(freq[i] != 0)
            freq[i] = count;
    }
    for(i = 0; i < n; i++)
    {
        if(freq[i] != 0)
            printf("%d occurs %d times\n", a[i], freq[i]);
    }
}
```

OUTPUT



```
5 10 10 20 30 20
10 occurs 2 times
20 occurs 2 times
30 occurs 1 times
```

10.Write a program to merge two arrays.

IPO

INPUT: Two arrays and sizes

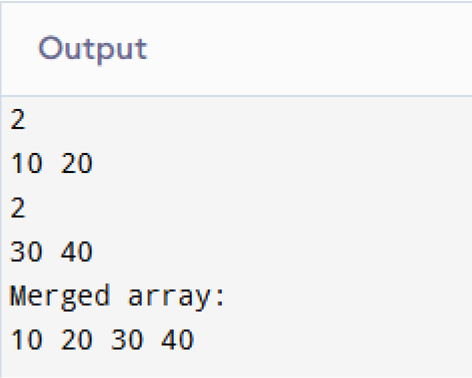
PROCESS: Copy both into one array

OUTPUT: Merged array

CODE;

```
#include <stdio.h>


void main()
{
    int a[50], b[50], merged[100];
    int n1, n2, i;
    scanf("%d", &n1);
    for(i = 0; i < n1; i++)
        scanf("%d", &a[i]);
    scanf("%d", &n2);
    for(i = 0; i < n2; i++)
        scanf("%d", &b[i]);
    for(i = 0; i < n1; i++)
        merged[i] = a[i];
    for(i = 0; i < n2; i++)
        merged[n1 + i] = b[i];
    printf("Merged array:\n");
    for(i = 0; i < n1 + n2; i++)
        printf("%d ", merged[i]);

}
```

Output;



```
2
10 20
2
30 40
Merged array:
10 20 30 40
```