1: Take 20 elements from user input and find its sum with the help of an array.

```
#include <iostream>

using namespace std;

int main()
{
    int numbers[20];
    int sum = 0;

    for (int i = 0; i < 20; i++)
    {
        cout << "Enter element " << i + 1 << ": ";
        cin >> numbers[i];
        sum += numbers[i];
    }

    cout << "The sum of the elements is: " << sum << endl;

    return 0;
}
```

2: Calculate the average of elements in an array of size 18.

```
#include <iostream>
#include <iomanip> // Include the <iomanip> library for fixed and
setprecision

using namespace std;

int main()
{
    int size = 18;
    int numbers[size];
    double sum = 0.0;
    for (int i = 0; i < size; i++)
    {
        cout << "Enter element " << i + 1 << ": ";
        cin >> numbers[i];
        sum += numbers[i];
    }

    double average = (sum) / size;

    // Set the output to display average with 2 decimal places
    cout << fixed << setprecision(2);

    cout << "The average of the elements is: " << average << endl;

    return 0;
}
```

3: Find the index of a specific element in an array, if the element is nor present, print -1. Ask the size of the array from the user and then implement it.

```
#include <iostream>

using namespace std;

int main()
{
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        cin >> arr[i];
    }

    int target;
    cout << "Enter the element to search for: ";
    cin >> target;

    int index = -1;

    for (int i = 0; i < size; i++)
    {
        if (arr[i] == target)
        {
            index = i;
            break;
        }
    }

    if (index != -1)
    {
        cout << "element present at " << index << endl;
    }
    else
    {
        cout << index << endl;
    }

    return 0;
}
```

4: Create an array of char types and store ‘a’ to ‘z’ in it. Then print the element of the arrays.

```
#include <iostream>
using namespace std;

int main()
{
    char charArray[26];
    for (int i = 0; i < 26; i++)
    {
        charArray[i] = 'a' + i;
    }

    for (int i = 0; i < 26; i++)
    {
        cout << charArray[i] << " ";
    }

    return 0;
}
```

5: Find the second largest element in an array of unique elements of size n. Where n>3.

```
int findSecondLargest(int arr[], int n)
{

    int largest = arr[0];
    int secondLargest = INT_MIN;

    for (int i = 1; i < n; i++)
    {
        if (arr[i] > largest)
        {
            secondLargest = largest;
            largest = arr[i];
        }
        else if (arr[i] > secondLargest && arr[i] != largest)
        {
            secondLargest = arr[i];
        }
    }

    return secondLargest;
}
```

6: Find the third smallest element in an array of unique elements size n. Where n>3.

```
int findThirdSmallest(int arr[], int n)
{
    int smallest = INT_MAX;
    int secondSmallest = INT_MAX;
    int thirdSmallest = INT_MAX;

    for (int i = 0; i < n; i++)
    {
        if (arr[i] < smallest)
        {
            thirdSmallest = secondSmallest;
            secondSmallest = smallest;
            smallest = arr[i];
        }
        else if (arr[i] < secondSmallest && arr[i] != smallest)
        {
            thirdSmallest = secondSmallest;
            secondSmallest = arr[i];
        }
        else if (arr[i] < thirdSmallest && arr[i] != smallest && arr[i] != secondSmallest)
        {
            thirdSmallest = arr[i];
        }
    }
    return thirdSmallest;
}
```

## What is Byte addressable?

https://www.tutorialspoint.com/difference-between-byte-addressable-memory-and-word-addressable-memory#:~:text=Byte%20addressable%20memory%20is%20one,it%20uses%20bytewise%20storage%20configuration.