Lecture 10

Array

Problem-1

Awesh is a software developer working on a project to manage a company's monthly sales data for a year. He needs to create a program to calculate the total sales for the year and the average sales per month.

Sample Data:

January: 1200
February: 1300
March: 1250
April: 1350
May: 1400
June: 1500
July: 1600
August: 1700
September: 1800
October: 1900
November: 2000
December: 2100

Output:

Total sales for the year: 19950
Average sales per month: 1662.5

Problem with this approach?

This approach is not scalable or efficient.
If more data needs to be added more code.
As programmers we need to think
efficiently.

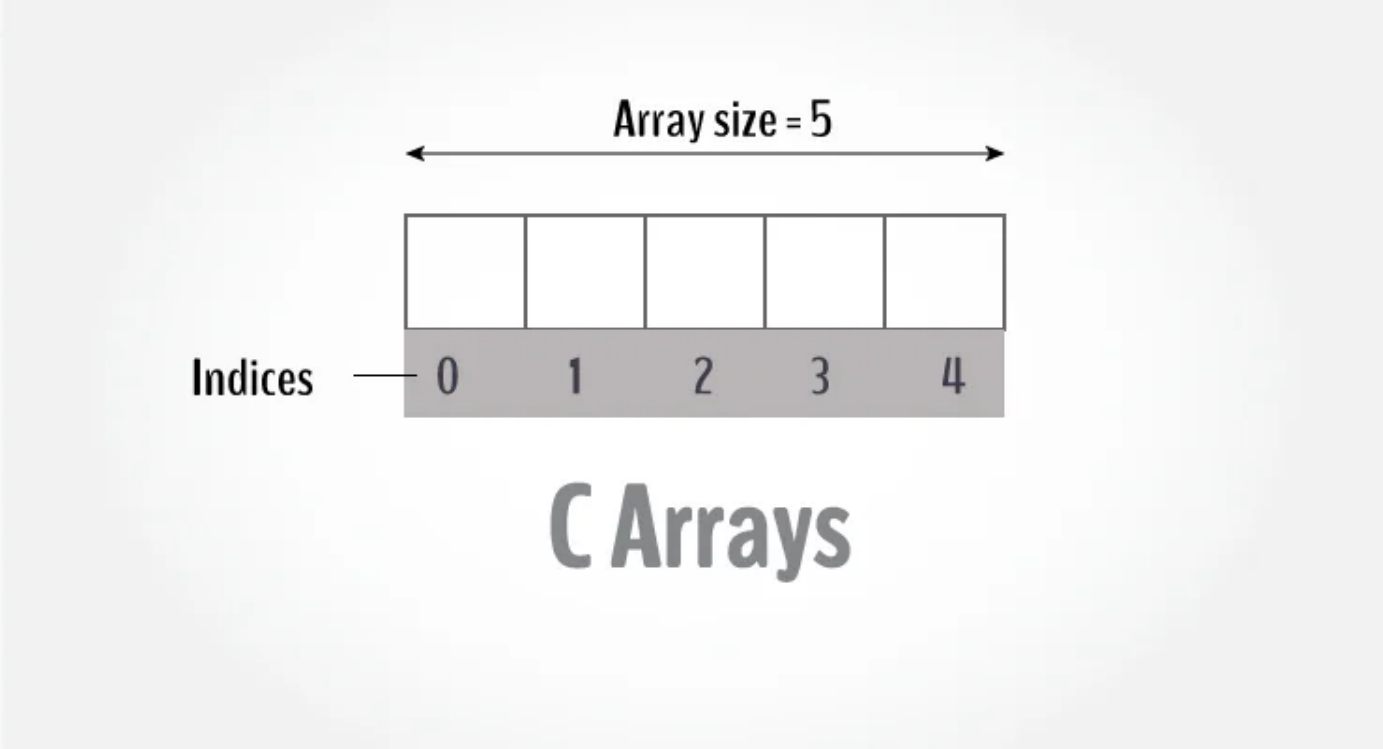
Solve

```c
#include <stdio.h>

int main()
{
    int January = 1200;
    int February = 1300;
    int March = 1250;
    int April = 1350;
    int May = 1400;
    int June = 1500;
    int July = 1600;
    int August = 1700;
    int September = 1800;
    int October = 1900;
    int November = 2000;
    int December = 2100;
    int total = January + February + March + April + May + June + July + August + September + October +
November + December;
    printf("Total Salary of the year is %d\n", total);
    double average = total / 12;
    printf("Average Salary is: %lf\n", average);
    return 0;
}
```

What is an Array?

1) A collection of elements of the same type stored in contiguous memory locations.
2) Elements are accessed by index.



Declaring and Initialising Array

Declaration:

dataType name[size];

```
int data[5];
```

Initialisation And Partial Initialisation:

```
int arr[5] = {1, 2, 3, 4, 5};
int arr[5] = {1, 2}; // Remaining initialized to 0
```

Access Array Elements

```c
#include <stdio.h>
int main()
{
    int ara[5] = {10, 20, 30, 40, 50};
    printf("First element: %d\n", ara[0]);
    printf("Third element: %d\n", ara[2]);
    return 0;
}
```

Modify Array Elements

```c
#include <stdio.h>
int main()
{
    int ara[10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    ara[1] = -1;
    printf("The 2nd element of array is: %d\n",ara[1]);
    return 0;
}
```

What happens if index not valid

```c
#include <stdio.h>
int main()
{
    int ara[5] = {6, 7, 4, 6, 9};
    printf("%d\n", ara[-1]);
    printf("%d\n", ara[5]);
    printf("%d\n", ara[100]);
    return 0;
}
```

Access Array By Looping

When the number of elements increases, it is not feasible to access them one by one. So we use loops:

```c
#include <stdio.h>
int main()
{
    int ara[10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int i;
    for(i = 0; i < 10; i++) {
        printf("%d th element is: %d\n", i+1, ara[i]);
    }
    return 0;
}
```

Let's Solve problem-1 with array

```c
#include <stdio.h>

int main()
{
    int monthlySales[12] =
{1200,1300,1250,1350,1400,1500,1600,1700,1800,1900,2000,2100};
    int total = 0;
    for(int i = 0;i < 12;i++){
        total += monthlySales[i];
    }
    printf("Total Salary of the year is %d\n", total);
    double average = (double)total / 12;
    printf("Average Salary is: %lf\n", average);
    return 0;
}
```

Problem-1 Modified

Awesh is a software developer working on a project to manage a company's monthly sales data for a year. He needs to create a program to calculate the total sales for the year and the average sales per month. But the company will provide the data as input.

Array Input

```c
// take input and store it in the 3rd element
scanf("%d", &mark[2]);

// take input and store it in the ith element
scanf("%d", &mark[i-1]);
```

Array Input

```c
// Program to take 5 values from the user and store them in an array
// Print the elements stored in the array

#include <stdio.h>

int main() {

  int values[5];

  printf("Enter 5 integers: ");

  // taking input and storing it in an array
  for(int i = 0; i < 5; ++i) {
     scanf("%d", &values[i]);
  }

  printf("Displaying integers: ");

  // printing elements of an array
  for(int i = 0; i < 5; ++i) {
     printf("%d\n", values[i]);
  }
  return 0;
}
```

Sum and Average of an array

```c
// Program to find the average of n numbers using arrays

#include <stdio.h>

int main() {

  int marks[10], i, n, sum = 0;
  double average;

  printf("Enter number of elements: ");
  scanf("%d", &n);

  for(i=0; i < n; ++i) {
    printf("Enter number%d: ",i+1);
    scanf("%d", &marks[i]);

    // adding integers entered by the user to the sum variable
    sum += marks[i];
  }

  // explicitly convert sum to double
  // then calculate average
  average = (double) sum / n;

  printf("Average = %.2lf", average);

  return 0;
}
```

Solution of Problem-1 Modified

SOLVE IT YOURSELF!!

Passing array into a function

```c
#include <stdio.h>

// Function to print the array
void printArray(int arr[], int size) {
    printf("Array elements: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

// Function to calculate the sum of the array
int sumArray(int arr[], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    return sum;
}
```

```c
int main() {
    int size;

    // Prompt the user to enter the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Declare the array with the given size
    int arr[size];

    // Get the array elements from the user
    printf("Enter %d elements for the array:\n",
size);for (int i = 0; i < size; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    // Print the array
    printArray(arr, size);

    // Calculate and print the sum of the array
    int totalSum = sumArray(arr, size);
    printf("Sum of array elements: %d\n", totalSum);

    return 0;
}
```

Reverse An array

```c
#include <stdio.h>
int main()
{
    int ara[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int ara2[10];
    int i, j;
    for(i = 0, j = 9; i < 10; i++, j--) {
        ara2[j] = ara[i];
    }
    for(i = 0; i < 10; i++) {
        ara[i] = ara2[i];
    }
    for(i = 0; i < 10; i++) {
        printf("%d\n", ara[i]);
    }
    return 0;
}
```

Swapping a variable

```c
#include <stdio.h>

int main() {
    // Step 1: Declare two integer variables
    int a = 10;
    int b = 20;

    // Print original values
    printf("Original values:\n");
    printf("a = %d, b = %d\n", a, b);

    // Step 2: Declare a temporary variable to hold one of the values during
swapint temp;

    // Step 3: Swap the values
    // Store 'a' in 'temp'
    temp = a;

    // Copy 'b' into 'a'
    a = b;

    // Copy 'temp' (original value of 'a') into 'b'
    b = temp;

    // Step 4: Print the swapped values
    printf("Swapped values:\n");
    printf("a = %d, b = %d\n", a, b);

    return 0;
}
```

Reverse An array

```c
#include <stdio.h>
int main()
{
    int ara[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int i, j, temp;
    for(i = 0, j = 9; i < 10; i++, j--) {
        temp = ara[j];
        ara[j] = ara[i];
        ara[i] = temp;
    }
    for(i = 0; i < 10; i++) {
        printf("%d\n", ara[i]);
    }
    return 0;
}
```

Reverse An array

```c
#include <stdio.h>
int main()
{
    int ara[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int i, j, temp;
    for(i = 0, j = 9; i < 5; i++, j--) {
        temp = ara[j];
        ara[j] = ara[i];
        ara[i] = temp;
    }
    for(i = 0; i < 10; i++) {
        printf("%d\n", ara[i]);
    }
    return 0;
}
```

Function For Array Reversal

```
void reverseArray(int arr[], int length)
{   for (int i = 0; i < length / 2; i++)
{       int temp = arr[i];
        arr[i] = arr[length - i - 1];
        arr[length - i - 1] = temp;
    }
}
```

Find Max Element

```c
#include <stdio.h>

int findMax(int arr[], int size) {
    // Step 1: Initialize a variable to hold the maximum value.
    // Start with the first element of the array.
    int max = arr[0];

    // Step 2: Loop through the rest of the array to find the maximum value.
    for (int i = 1; i < size; i++) {
        // If the current array element is greater than the current maximum,
        // update the maximum to be the current element.
        if (arr[i] > max) {
            max = arr[i];
        }
    }

    // Step 3: Return the maximum value found.
    return max;
}
```

```c
int main() {
    // Example array
    int arr[] = {3, 5, 7, 2, 8, 10, 6, 12, 9};

    // Calculate the size of the array
    int size = sizeof(arr) / sizeof(arr[0]);

    // Call the function to find the maximum value
    int maxValue = findMax(arr, size);

    // Print the maximum value
    printf("The maximum value in the array is: %d\n",
maxValue);
    return 0;
}
```

Find Minimum Element

SOLVE IT YOURSELF!!

Find Duplicates in an array

```c
// Function to find and print duplicates in an array
void findDuplicates(int arr[], int size) {
    // Array to mark whether an element has already been identified as a duplicate
    int found[size];

    // Initialize the found array to zero (indicating no duplicates found
initfor(int i = 0; i < size; i++) {
        found[i] = 0;
    }

    printf("Duplicate elements: ");
    int has_duplicates = 0; // Flag to check if any duplicates are found

    // Iterate through each element in the array
    for (int i = 0; i < size; i++) {
        if (found[i] == 1) {
            // If already marked as a duplicate, skip
            continue;
        }

        int is_duplicate = 0; // Flag to check if current element is a duplicate

        // Compare with the rest of the array
        for (int j = i + 1; j < size; j++) {
            if (arr[i] == arr[j]) {
                found[j] = 1; // Mark as duplicate
                is_duplicate = 1;
            }
        }

        if (is_duplicate) {
            printf("%d ", arr[i]); // Output the duplicate element
            has_duplicates = 1;
        }
    }

    if (has_duplicates == 0) {
        printf("None"); // If no duplicates are found
    }

    printf("\n");
}
```