CS1160
Lab 6: Arrays

I. Arrays

An array is a data structure that can store a collection of elements of the **same type**. In other words, an array is a variable that can store multiple values.

- Array declaration:

```
Data_type array_name [array_size];
```

Example:

```
int numbers[4];
```

- Initializing Arrays: done in two different ways as follows:

```
numbers[0]=8;
numbers[1]=12;
numbers[2]=16;
numbers[3]=20;

OR
numbers[4] = {8,12,16,20};
```

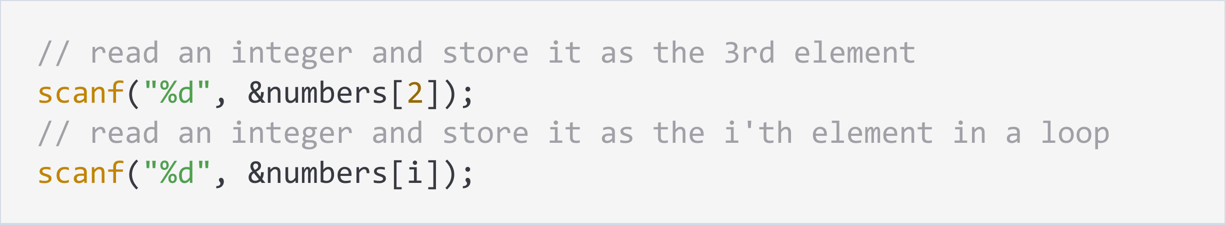
- Access Array Elements: by placing the index of the element within square brackets after the name of the array. :

Example:

```
int x = numbers[4]; // copying the value of the fourth/last
element in the array to the variable x
```

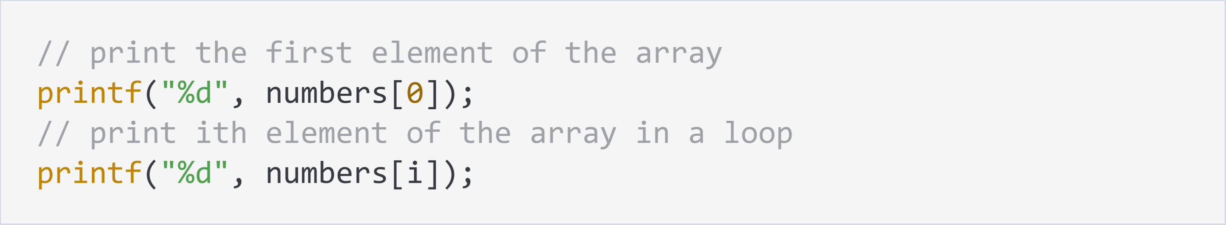
II. Input and Output Array Elements

To read / input values to the array simply use the **scanf** function



```
// read an integer and store it as the 3rd element
scanf("%d", &numbers[2]);
// read an integer and store it as the i'th element in a loop
scanf("%d", &numbers[i]);
```

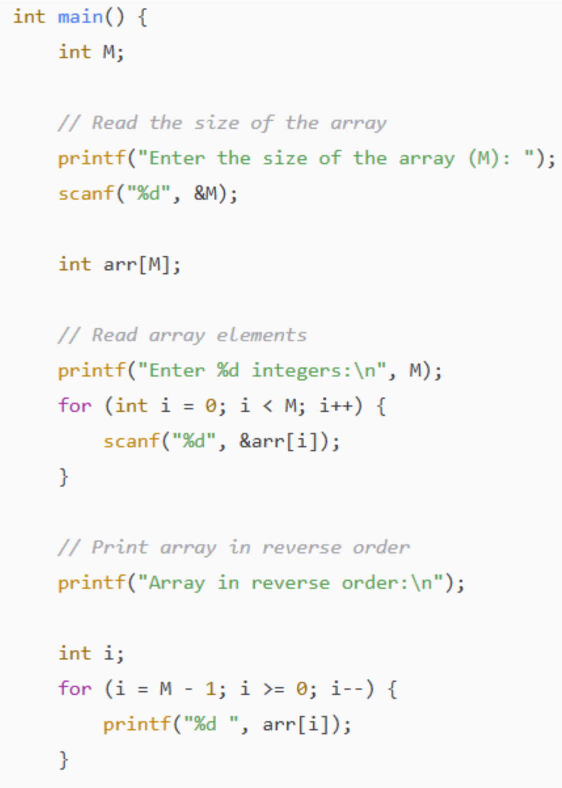
To print the array values simply use the **printf** function



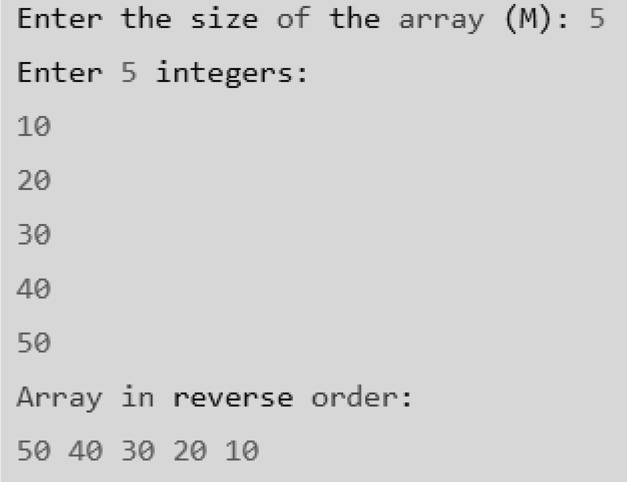
```
// print the first element of the array
printf("%d", numbers[0]);
// print ith element of the array in a loop
printf("%d", numbers[i]);
```

III. Example

1. Write a C program that **reads in** an **integer array of size M, M is entered by the user**. It will then **print** the array in reverse order.

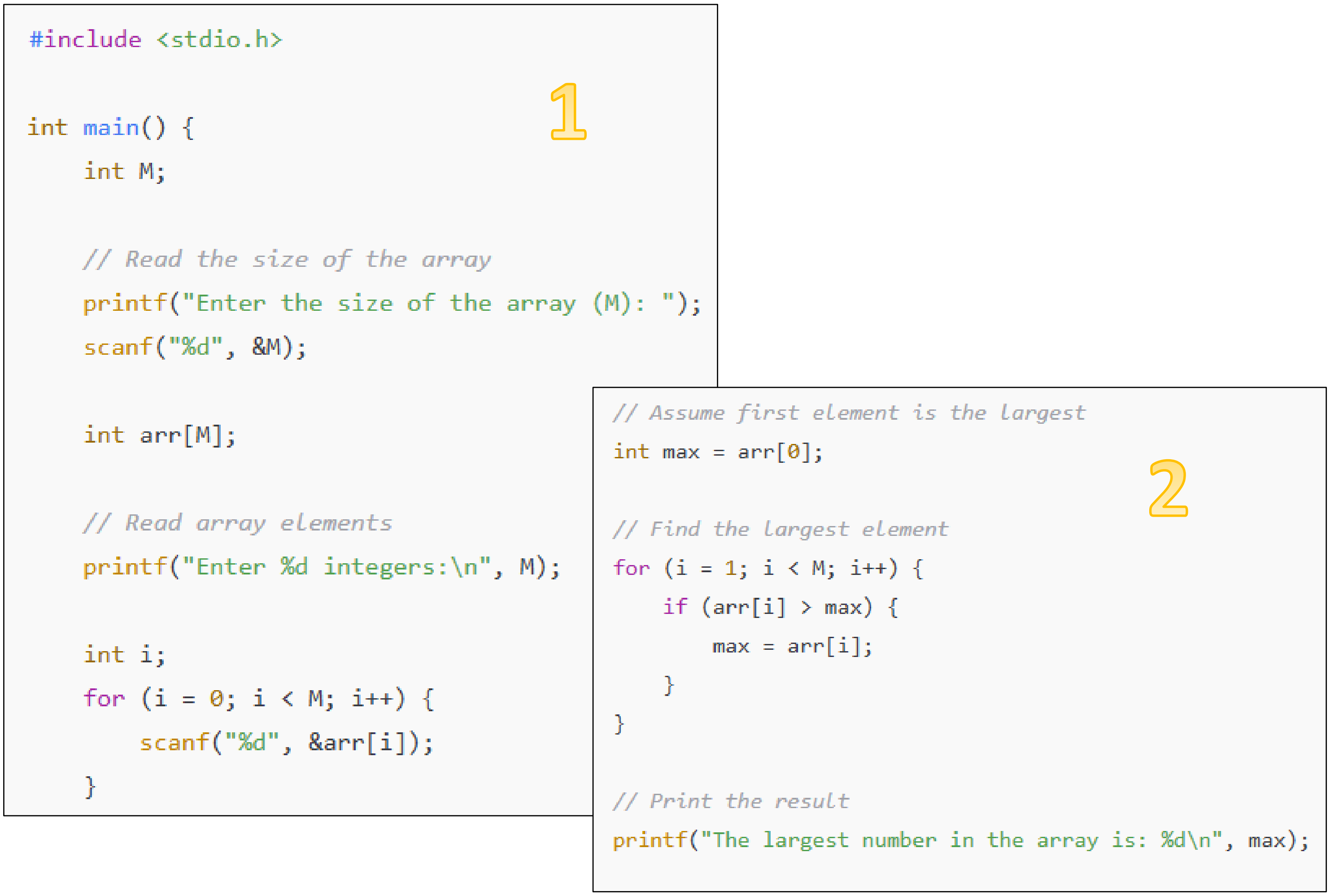


```
int main() {
    int M;

    // Read the size of the array
    printf("Enter the size of the array (M): ");
    scanf("%d", &M);

    int arr[M];

    // Read array elements
    printf("Enter %d integers:\n", M);
    for (int i = 0; i < M; i++) {
        scanf("%d", &arr[i]);
    }

    // Print array in reverse order
    printf("Array in reverse order:\n");

    int i;
    for (i = M - 1; i >= 0; i--) {
        printf("%d ", arr[i]);
    }
```



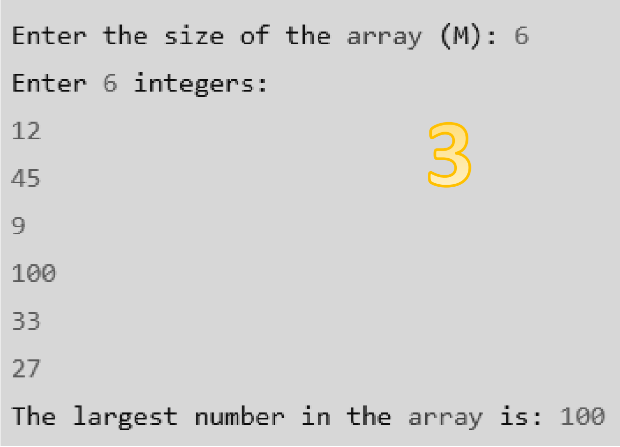
```
Enter the size of the array (M): 5
Enter 5 integers:
10
20
30
40
50
Array in reverse order:
50 40 30 20 10
```

2. Write a C program that reads an array of **M integers**, then finds and prints the **largest number** in the array.



```
#include <stdio.h>

int main() {
    int M;

    // Read the size of the array
    printf("Enter the size of the array (M): ");
    scanf("%d", &M);

    int arr[M];

    // Read array elements
    printf("Enter %d integers:\n", M);

    int i;
    for (i = 0; i < M; i++) {
        scanf("%d", &arr[i]);
    }
```

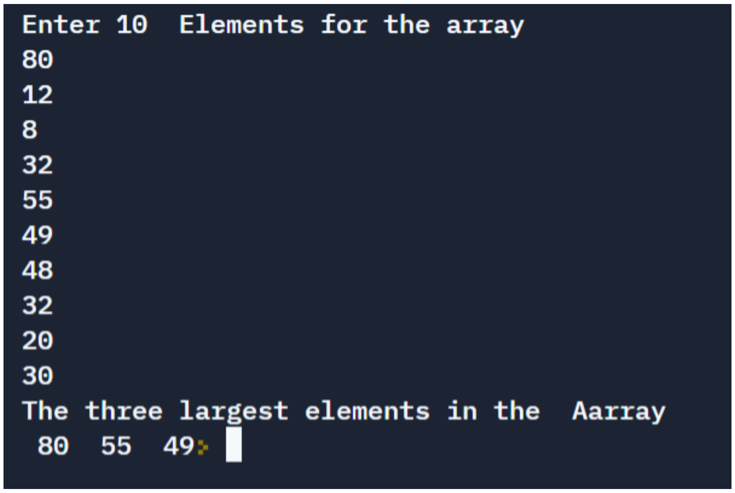
```
// Assume first element is the largest
int max = arr[0];

// Find the largest element
for (i = 1; i < M; i++) {
    if (arr[i] > max) {
        max = arr[i];
    }
}

// Print the result
printf("The largest number in the array is: %d\n", max);
```



```
Enter the size of the array (M): 6
Enter 6 integers:
12
45
9
100
33
27
The largest number in the array is: 100
```

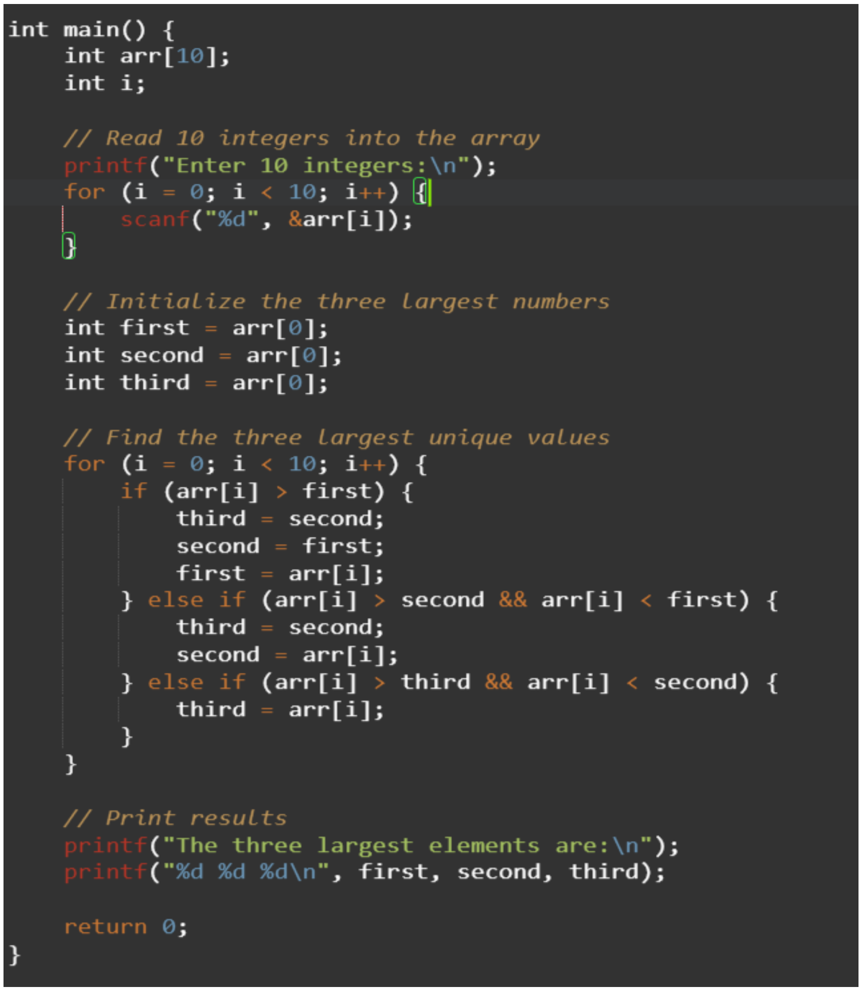
3. Write a program in C to find the three largest elements in an integer array of size **10**.

Sample Output



```
Enter 10  Elements for the array
80
12
8
32
55
49
48
32
20
30
The three largest elements in the  Aarray
 80  55  49:
```

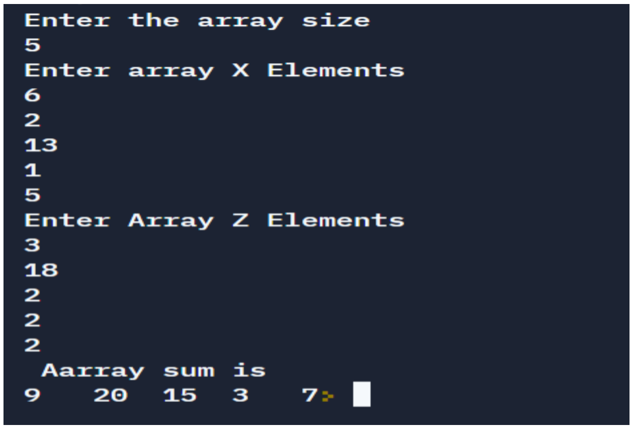
Solution:



```c
int main() {
    int arr[10];
    int i;

    // Read 10 integers into the array
    printf("Enter 10 integers:\n");
    for (i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }

    // Initialize the three largest numbers
    int first = arr[0];
    int second = arr[0];
    int third = arr[0];

    // Find the three largest unique values
    for (i = 0; i < 10; i++) {
        if (arr[i] > first) {
            third = second;
            second = first;
            first = arr[i];
        } else if (arr[i] > second && arr[i] < first) {
            third = second;
            second = arr[i];
        } else if (arr[i] > third && arr[i] < second) {
            third = arr[i];
        }
    }

    // Print results
    printf("The three largest elements are:\n");
    printf("%d %d %d\n", first, second, third);

    return 0;
}
```

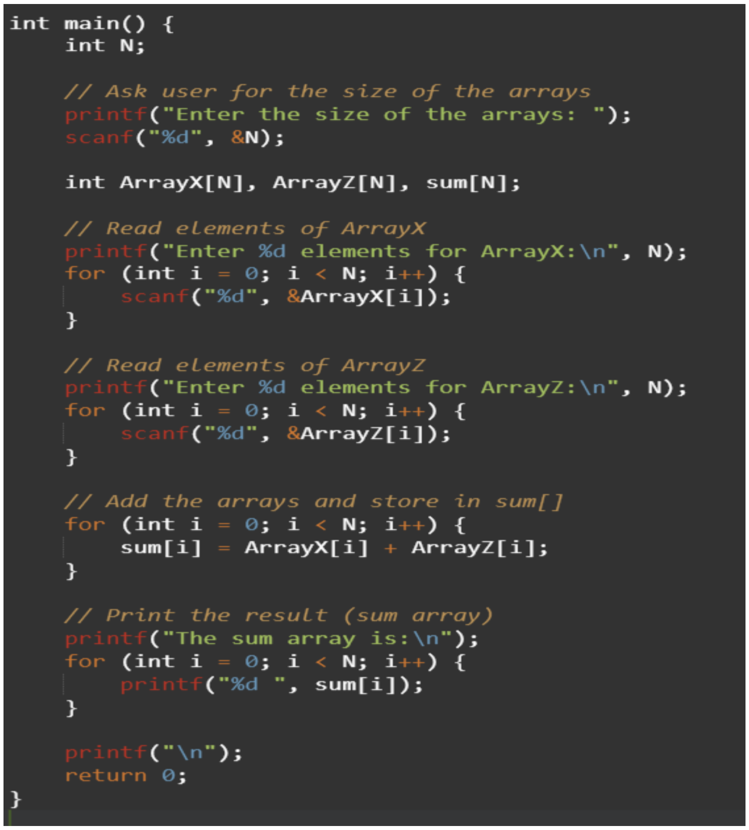
4. Write a C program that:
    a. Declares two integer arrays, **ArrayX** and **ArrayZ**, both size **N**.
    b. The programme should read elements to **ArrayX**, and **ArrayZ**.
    c. It will then **add** the two arrays and save the result in an **array called “sum”** .
    d. Print the array “sum”.

Sample Output



```
Enter the array size
5
Enter array X Elements
6
2
13
1
5
Enter Array Z Elements
3
18
2
2
2
 Aarray sum is
9    20   15   3    7>
```

Solution:



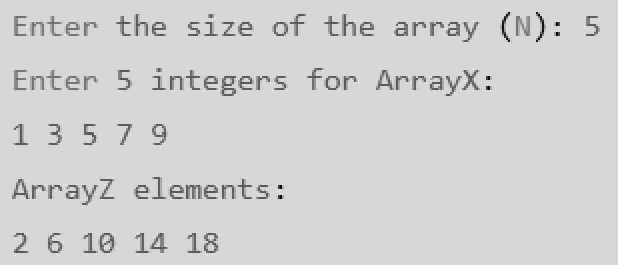
```c
int main() {
    int N;

    // Ask user for the size of the arrays
    printf("Enter the size of the arrays: ");
    scanf("%d", &N);

    int ArrayX[N], ArrayZ[N], sum[N];

    // Read elements of ArrayX
    printf("Enter %d elements for ArrayX:\n", N);
    for (int i = 0; i < N; i++) {
        scanf("%d", &ArrayX[i]);
    }

    // Read elements of ArrayZ
    printf("Enter %d elements for ArrayZ:\n", N);
    for (int i = 0; i < N; i++) {
        scanf("%d", &ArrayZ[i]);
    }

    // Add the arrays and store in sum[]
    for (int i = 0; i < N; i++) {
        sum[i] = ArrayX[i] + ArrayZ[i];
    }

    // Print the result (sum array)
    printf("The sum array is:\n");
    for (int i = 0; i < N; i++) {
        printf("%d ", sum[i]);
    }

    printf("\n");
    return 0;
}
```

IV. Tasks

1. Write a C program that reads **M integers** into an array (where M is entered by the user). The program should then:
   - Reads **M elements** into array **M**
   - **print only the even** elements in the array.
   - Count the number of **even** elements

2. Write a C program that:
   - Declares an integer array ArrayX of size N.
   - Reads N **elements** into array ArrayX.
   - Creates another array ArrayZ of the same size, where each element of ArrayZ is ArrayX[i] × 2.
   - Prints array ArrayZ.

Sample Output



```
Enter the size of the array (N): 5
Enter 5 integers for ArrayX:
1 3 5 7 9
ArrayZ elements:
2 6 10 14 18
```

3. Write a program in C to find the three minimum elements in an integer array of size **10**.