Arrays

1) Write a C program to access an array element

Access an Array Element

Algorithm

1. Declare and initialize an array.
2. Access a specific element using its index.
3. Print the value.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {10, 20, 30, 40, 50};
   int index = 2; // Access the 3rd element (index starts from 0)
   printf("Element at index %d: %d\n", index, numbers[index]);
   return 0;
}
```

2) Write a C program to change the value of a specific element

Change the Value of a Specific Element

Algorithm

1. Declare and initialize an array.
2. Change a specific index value.
3. Print the modified array.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {10, 20, 30, 40, 50};
   int index = 2; // Modify the 3rd element
   numbers[index] = 100; // Change value
   printf("Updated array: ");
   for (int i = 0; i < 5; i++) {
      printf("%d ", numbers[i]);
   }
   return 0;
}
```

3) Write a C program to loop through an array

Loop Through an Array

Algorithm

1. Declare and initialize an array.
2. Use a loop to iterate through each element.
3. Print each value.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {5, 10, 15, 20, 25};
   printf("Array elements: ");
   for (int i = 0; i < 5; i++) {
      printf("%d ", numbers[i]);
   }
   return 0;
}
```

4) Write a C program to create a specific size and adding the elements later

Create a Specific Size Array and Add Elements Later

Algorithm

1. Declare an array with a specific size.
2. Take user input to fill the array.
3. Print the elements.

C Code

```
#include <stdio.h>

int main() {
   int numbers[5]; // Declare an array of size 5
   printf("Enter 5 numbers: ");
   for (int i = 0; i < 5; i++) {
      scanf("%d", &numbers[i]);
   }
   printf("You entered: ");
   for (int i = 0; i < 5; i++) {
      printf("%d ", numbers[i]);
   }
   return 0;
}
```

5) Write a C program to print the size of the array

Print the Size of the Array

Algorithm

1. Declare an array.
2. Use sizeof() to find the size in bytes.
3. Print the size.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {1, 2, 3, 4, 5};
   printf("Size of the array: %lu bytes\n", sizeof(numbers));
   return 0;
}
```

6) Write a C program to find the length of an array

Find the Length of an Array

Algorithm

1. Find the size using sizeof().
2. Divide by sizeof() of an element.
3. Print the length.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {10, 20, 30, 40, 50};
   int length = sizeof(numbers) / sizeof(numbers[0]);
   printf("Length of the array: %d\n", length);
   return 0;
}
```

7) Create a program that calculates the average of different ages

Calculate the Average of Different Ages

Algorithm

1. Declare an array and initialize ages.
2. Sum the elements.
3. Divide by the total count.
4. Print the average.

C Code

```
#include <stdio.h>

int main() {
   int ages[] = {25, 30, 35, 40, 45};
   int sum = 0, count = sizeof(ages) / sizeof(ages[0]);
   for (int i = 0; i < count; i++) {
      sum += ages[i];
   }
   float average = (float)sum / count;
   printf("Average age: %.2f\n", average);
   return 0;
}
```

8) Write a C program that finds the lowest age among different ages

Find the Lowest Age in an Array

Algorithm

1. Declare an array of ages.
2. Assume the first element is the smallest.
3. Compare with other elements and update the smallest.
4. Print the smallest value.

C Code

```
#include <stdio.h>

int main() {
   int ages[] = {25, 30, 18, 40, 22};
   int min = ages[0];
   for (int i = 1; i < 5; i++) {
      if (ages[i] < min) {
         min = ages[i];
      }
   }
   printf("Lowest age: %d\n", min);
   return 0;
}
```

9) Write a C program to reverse the elements of an array.

Reverse the Elements of an Array

Algorithm

1. Declare an array.
2. Swap elements from start and end moving toward the center.
3. Print the reversed array.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {1, 2, 3, 4, 5};
   int length = sizeof(numbers) / sizeof(numbers[0]);
   for (int i = 0; i < length / 2; i++) {
      int temp = numbers[i];
      numbers[i] = numbers[length - i - 1];
      numbers[length - i - 1] = temp;
   }
   printf("Reversed array: ");
   for (int i = 0; i < length; i++) {
      printf("%d ", numbers[i]);
   }
   return 0;
}
```

10) Write a C program to find the sum of all elements in an array.

Find the Sum of All Elements in an Array

Algorithm

1. Declare an array.
2. Initialize sum = 0.
3. Loop through and add each element to sum.
4. Print the total sum.

C Code

```
#include <stdio.h>

int main() {
   int numbers[] = {5, 10, 15, 20, 25};
   int sum = 0;
   for (int i = 0; i < 5; i++) {
      sum += numbers[i];
   }
   printf("Sum of elements: %d\n", sum);
   return 0;
}
```