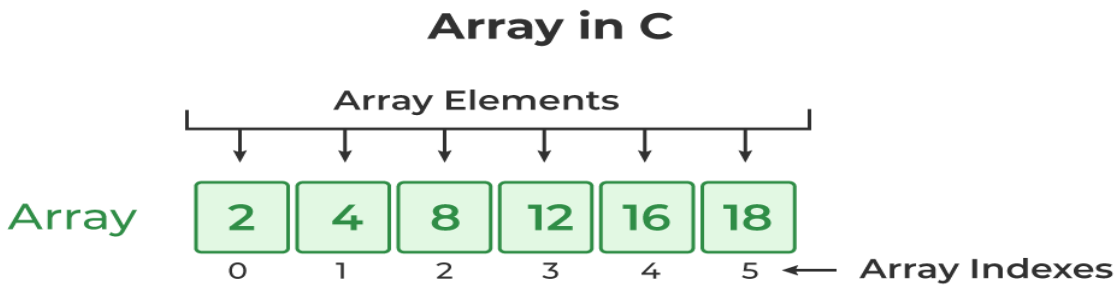
ARRAYS AND STRINGS

1D and 2D Arrays in C

Definition:

An array is a collection of elements of the same data type stored in contiguous memory locations. Arrays in C are used to store multiple values of the same type, and they are defined in various dimensions. A 1D array holds elements in a single row, while a 2D array represents elements in a grid format.



1D Array:

A 1D array is a linear sequence of elements that can be accessed using a single index.

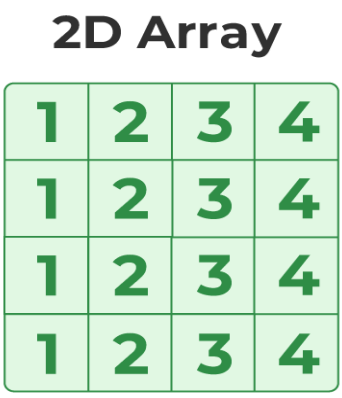
Example of Declaration:

```
int array[6];
```

Program:

```
#include <stdio.h>

int main() {
   int array[6], i;
   printf("Enter 6 elements for the array: \n");
   for(i = 0; i < 6; i++) {
      scanf("%d", &array[i]);
   }
   printf("The elements in the array are: ");
   for(i = 0; i < 6; i++) {
      printf("%d ", array[i]);
   }
   return 0;
}
```



2D Array:

A 2D array is a collection of elements arranged in rows and columns.

Example of Declaration:

int matrix[4][4]; // Declares a 4x4 matrix of integers

Program:

```
#include <stdio.h>
int main() {
   int matrix[2][2], i, j;
   printf("Enter 4 elements for the 2x2 matrix: \n");
   for(i = 0; i < 2; i++) {
      for(j = 0; j < 2; j++) {
         scanf("%d", &matrix[i][j]);  // Input elements
      }
   }
   printf("The 2x2 matrix is:\n");
   for(i = 0; i < 2; i++) {
      for(j = 0; j < 2; j++) {
         printf("%d ", matrix[i][j]);  // Output matrix
      }
      printf("\n");
   }
   return 0;
}
```

Strings in C

Definition:

A string in C is a sequence of characters that ends with a null character (\0). Strings are essentially character arrays, and string handling functions are used to manipulate them.

String Handling Functions

1. strlen(): This function calculates the length of a string excluding the null terminator.

2. strcpy(): It copies one string into another.

3. strcmp(): It compares two strings lexicographically.

4. strcat(): This function appends one string to the end of another.

5. strrev(): Though not a standard function in C, it can be implemented to reverse a string.

Program for strlen()

```
#include <stdio.h>

#include <string.h>

int main() {

   char str[100];


   printf("Enter a string: ");

   fgets(str, 100, stdin);  // Input string

   printf("The length of the string is: %ld\n", strlen(str) - 1);  // Calculate and display length

   return 0;

}
```

Program for strcpy()

```
#include <stdio.h>

#include <string.h>

int main() {

   char source[100], destination[100];

   printf("Enter the source string: ");

   fgets(source, 100, stdin);  // Input source string

   strcpy(destination, source);  // Copy string

   printf("The copied string is: %s", destination);  // Display copied string

   return 0;
```

```
}
```

Program for strcmp()

```
#include <stdio.h>
#include <string.h>
int main() {
   char str1[100], str2[100];
   printf("Enter the first string: ");
   fgets(str1, 100, stdin);  // Input first string
   printf("Enter the second string: ");
   fgets(str2, 100, stdin);  // Input second string
   int result = strcmp(str1, str2);  // Compare strings
   if(result == 0) {
      printf("The strings are identical.\n");
   } else if(result > 0) {
      printf("The first string is lexicographically greater.\n");
   } else {
      printf("The first string is lexicographically smaller.\n");
   }
   return 0;
}
```

Program for strcat()

```
#include <stdio.h>
#include <string.h>
int main() {
   char str1[100], str2[100];
   printf("Enter the first string: ");
   fgets(str1, 100, stdin);  // Input first string
   printf("Enter the second string: ");
   fgets(str2, 100, stdin);  // Input second string
   strcat(str1, str2);  // Concatenate str2 to str1
   printf("The concatenated string is: %s", str1);  // Display result
```

```
    return 0;
}
```

Program for Custom strrev()

```
#include <stdio.h>
#include <string.h>
void reverseString(char* str) {
    int start = 0;
    int end = strlen(str) - 1;
    char temp;
    while (start < end) {
        // Swap characters
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
        start++;
        end--;
    }
}
int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, 100, stdin);
    reverseString(str);
    printf("The reversed string is: %s", str);
    return 0;
}
```