C Programming

Functions

Passing a 1-D array to a function

An entire 1-D array may be passed to a function by using only the name of the array as the parameter when calling the function.

The following program passes an integer array to a function, which calculates the sum of the array…

```
/*
This program passes a 1-D array to a function, which calculates the sum
of the array and display to standard output (screen)
*/

#include <stdio.h>

#define SIZE 5

// Function signature
int sum_array(int[]);

int main()
{
   int values[SIZE];
   int i;
   int sum = 0;


   printf("Enter %d numbers\n", SIZE);

   //Enter data into the array
   for(i = 0; i < SIZE; i++)
   {
       scanf("%d", & values[i]); // scanf("%d", & *(values + i));

   } // end for
```

```
    // Calculate the sum of the contents of the array
    sum = sum_array(values);

    // Display the sum of the contents of the array
    printf("\nThe sum of the array is %d", sum);

    return 0;

} // end main()


/* Function sum_array() will calculate the sum of the contents of any
1-D array
*/
int sum_array(int my_array[])
{
    int total = 0;
    int i;

    // calculate the total of the array
    for(i = 0; i < SIZE; i++)
    {
        total = total + my_array[i];

    } // end for

    return total;

} // end sum_array()
```

Repl 15.1: https://replit.com/@michaelTUDublin/151-Passing-1-D-array-to-a-function-v1

As you remember, in C the name of an array is equivalent to the address location of the 1st element of the array. Therefore, when you pass a 1-D array to a function, the function parameter can be a pointer variable that will grab the address of the 1st element.

Modifying the program above to do this is as follows:

```c
/*
This program passes a 1-D array to a function, which calculates the sum
of the array and display to standard output (screen)
*/

#include <stdio.h>

#define SIZE 5

// Function signature
int sum_array(int *);

int main()
{
   int values[SIZE];
   int i;
   int sum = 0;


   printf("Enter %d numbers\n", SIZE);

   //Enter data into the array
   for(i = 0; i < SIZE; i++)
   {
       scanf("%d", & values[i]); // scanf("%d", & *(values + i));

   } // end for

   // Calculate the sum of the contents of the array
   sum = sum_array(values);

   // Display the sum of the contents of the array
   printf("\nThe sum of the array is %d", sum);

   return 0;

} // end main()
```

```
/* Function sum_array() will calculate the sum of the contents of any
1-D array
*/
int sum_array(int *my_array)
{
   int total = 0;
   int i;

   // calculate the total of the array
   for(i = 0; i < SIZE; i++)
   {
        total = total + *(my_array + i);

   } // end for

   return total;

} // end sum_array()
```

Repl 15.2: https://replit.com/@michaelTUDublin/152-Passing-1-D-array-to-a-function-v2

NOTE: Passing a 1-D array as a parameter to a function in C is always Pass by Reference. This is because in C, the name of an array is equivalent to the address of the 1st element of that array and when you pass the array name in the function call, you are actually passing the address location of the 1st element.

Therefore, if you make any changes to your parameter array in your function, you will be changing the original array back where the function was called.

Passing a 2-D array to a function

Passing a 2-D array to a function is very similar to passing a 1-D array. However, there is a small change needed in the code.

When you pass a 2-D array, the function signature requires you to explicitly state the number of Columns. For example, let's take the example above and change it so that it uses a 2-D array and passes it to the function to calculate its sum:

```c
/*
This program passes a 2-D array to a function, which calculates the sum
of the array and display to standard output (screen)
*/
#include <stdio.h>

#define ROW 2
#define COL 3

// Function signature
int sum_array(int [][COL]);

int main()
{
  int values[ROW][COL];
  int i,j;
  int sum = 0;

  printf("Enter %d numbers\n", ROW*COL);

  //Enter data into the array
  for(i = 0; i < ROW; i++)
  {
      for(j = 0; j < COL; j++)
      {
           scanf("%d", & values[i][j]);

      } // end inner for

  } // end outer for

  // Calculate the sum of the contents of the array
  sum = sum_array(values);

  // Display the sum of the contents of the array
  printf("\nThe sum of the array is %d\n", sum);

  return 0;
```

```
} // end main()


/* Function sum_array() will calculate the sum of the contents of any
2-D array
*/
int sum_array(int my_array[][COL])
{
  int total = 0;
  int i,j;


  // calculate the total of the array
  for(i = 0; i < ROW; i++)
  {
      for(j = 0; j < COL; j++)
      {
           total = total + my_array[i][j];

      } // end inner for
  } // end outer for

  return total;

} // end sum_array()
```

Repl 15.3: https://replit.com/@michaelTUDublin/153-Passing-2-D-array-to-a-function