

Passing pointers to functions in C

C programming allows passing a pointer to a function. To do so, simply declare the function parameter as a pointer type.

Following is a simple example where we pass an unsigned long pointer to a function and change the value inside the function which reflects back in the calling function –

```
#include <stdio.h>
#include <time.h>

void getSeconds(unsigned long *par);

int main () {

    unsigned long sec;
    getSeconds( &sec );

    /* print the actual value */
    printf("Number of seconds: %ld\n", sec );

    return 0;
}

void getSeconds(unsigned long *par) {
    /* get the current number of seconds */
    *par = time( NULL );
    return;
}
```

[Live Demo](#)

When the above code is compiled and executed, it produces the following result –

```
Number of seconds :1294450468
```

The function, which can accept a pointer, can also accept an array as shown in the following example –

[Live Demo](#)

```
#include <stdio.h>

/* function declaration */
double getAverage(int *arr, int size);

int main () {

    /* an int array with 5 elements */
    int balance[5] = {1000, 2, 3, 17, 50};
    double avg;

    /* pass pointer to the array as an argument */
    avg = getAverage( balance, 5 ) ;

    /* output the returned value */
    printf("Average value is: %f\n", avg );
    return 0;
}

double getAverage(int *arr, int size) {

    int i, sum = 0;
    double avg;

    for (i = 0; i < size; ++i) {
        sum += arr[i];
    }

    avg = (double)sum / size;
    return avg;
}
```

When the above code is compiled together and executed, it produces the following result –

Average value is: 214.40000