

Returning Data in a C Function

Now that you have seen how data can be passed to functions for them to perform operations on, the next level of function is one which can send back a result. These “Level Three” functions can receive any number of arguments but can return a single result only. This is essentially what the “return 0” in our main() function has been doing in all of our programs, but at this point, the result is generally something that is required in code for further operation. We have seen many examples of pre-existing Level Three functions; pow(), strlen(), abs(), floor() etc are all functions which are passed data and return an appropriate result.

The example below shows a Level Three function and it's use.

```
#include <stdio.h>

//Function Prototypes
int numSum(int, int);           //A function to add two integers

int main (void)
{
    int num1=5, num2=3;        //Two integers to add

    //Passing the values to the function. Note the result is sent directly to the printf
    printf("The sum of %i and %i is %i",num1, num2, numSum(num1,num2));

    system("PAUSE");
    return 0;
}

/*****
Function:    numSum
Does:       Adds two integers
Receives:   Two integers
Uses:       Nothing
Returns:    The sum of the integers as an integer
*****/
int biggerNum (int x, int y)
{
    int sum;                   //This could have been done simply as one
    sum = x + y;               //line saying "return x + y"
    return sum;
}
```

In the example above, the result is just printed out directly and is not stored in the main() function. Often it is desirable to store a returned result with a variable which is local to main(). This can be done easily with a statement like “ answer = numSum(num1,num2)”. There are many ways which this can be used effectively in your programs.

Assignment

Note: Some of these can be adapted from other programs you have already done.

1. Write a properly structured C program which has a user enter two integers. The program will then call two functions LCM and GCF which will be sent the two numbers and will return the appropriate result for each. The results should both be stored locally in main() and then displayed in main in a single printf along with the original values.
2. Write a properly structured C program which allows a user to enter a number in main() as a string for temporary storage. Then call the function testNum which will be sent this string and will return an integer 0 if it is NOT an integer, and a 1 if it could be an integer.
3. Write a properly structured C program which uses a menu() function called from main(), which displays a menu of program options as integer choices. Have the function return the integer to main if it is appropriate.
4. Write a properly structured C program which allows a user to enter their percentage mark as a float in main(). Have the function letterGrade be sent this mark and have it return a letter grade (A, B, C, D, F) as appropriate.
5. Write a properly structured C program which allows a user to enter a string and a letter in main(). Call the function letterCount which will be sent the string and the letter and will count the number of occurrences of that letter (upper and lower case) and will return that result to main().
- 6.