# Python Addition Calculator

## Create the script files

1. Create a folder called py_calc and change directory

```
pi@raspberrypi:~ $ mkdir py_calc
pi@raspberrypi:~ $ cd py_calc/
pi@raspberrypi:~/py_calc $
```

2. Create the source file for the program

```
pi@raspberrypi:~/asm_calc $ nano adder.py
```

3. Write the following code in the source file

```
  GNU nano 2.7.4                    File: adder.py

a = input("Enter a number: ")
b = input("Enter another number: ")
c = int(a) + int(b)
print("{0} + {1} = {2}".format(a, b, c))
```

4. Press ctrl + o to write out the file
5. Press ctrl + x to return to the command line
6. Run the program.

```
pi@raspberrypi:~/py_calc $ python adder.py
Enter a number: 89
Enter another number: 22
89 + 22 = 111
pi@raspberrypi:~/py_calc $
```

## Suggestions

Make changes to the source file and run the program again.

A. Create a syntax error on the last line (e.g. remove a quotation mark from the last line).

B. Change the program to subtract b from a, and display the result.

# C Addition Calculator

## Create the source code

1. Create a folder called c_calc and change directory

```
pi@raspberrypi:~ $ mkdir c_calc
pi@raspberrypi:~ $ cd c_calc/
pi@raspberrypi:~/c_calc $
```

2. Create the source file for the program

```
pi@raspberrypi:~/c_calc $ nano adder.c
```

3. Write the following code to the source file

```
  GNU nano 2.7.4                    File: adder.c                    Modified

#include <stdio.h>

int main()
{
        int a;
        int b;

        printf("Enter a number: ");
        scanf("%d", &a);
        printf("Enter another number: ");
        scanf("%d", &b);

        int c = a + b;

        printf("%d + %d = %d \n", a, b, c);
}
```

4. Press ctrl + o to write out the file
5. Press ctrl + x to return to the command line

## Compile the source file

6. Run gcc to compile the source file (adder.c) into object code (adder).

```
pi@raspberrypi:~/c_calc $ gcc adder.c -o adder
```

7. List the contents of the directory, to the source code has been compiled.

```
pi@raspberrypi:~/c_calc $ ls
adder  adder.c
```

8. Run the program (object code)

```
pi@raspberrypi:~/c_calc $ ./adder
Enter a number: 45
Enter another number: 8
45 + 8 = 53
pi@raspberrypi:~/c_calc $
```

## Suggestions

Make changes to the source file and run the program again.

A. Create a syntax error on the last line (e.g. remove a quotation mark from the last line).

B. Change the program to subtract b from a, and display the result.

# Assembly Language Addition Calculator

NOTE: This first assembly program does not accept user input. The values are loaded into the register in the instructions.

## Create the source code

1. Create a folder called as_calc and change directory

```
pi@raspberrypi:~ $ mkdir asm_calc
pi@raspberrypi:~ $ cd asm_calc/
pi@raspberrypi:~/asm_calc $
```

2. Create the source file for the program

```
pi@raspberrypi:~/asm_calc $ nano adder1.s
```

3. Write the following code to the source file

```
 GNU nano 2.7.4                    File: adder1.s

.data
        string: .asciz "%d + %d = %d \n"
.text
.global main

main:
        PUSH {ip, lr}
        LDR R0, =string
        MOV R1, #65
        MOV R2, #44
        ADD R3, R1, R2
        BL printf
        POP {ip, pc}
```

4. Press ctrl + o to write out the file
5. Press ctrl + x to return to the command line

## Assemble the source file

6. Assemble the source file (adder1.s) then compile the translated assembly to object code (adder1).

```
pi@raspberrypi:~/dev/asm_calc $ as adder1.s -o adder1.a
pi@raspberrypi:~/dev/asm_calc $ gcc adder1.a -o adder1
```

7. Run the program (object code)

```
pi@raspberrypi:~/asm_calc $ ./adder1
65 + 44 = 109
pi@raspberrypi:~/asm_calc $
```

## Suggestions

Make changes to the source file, compile and run the program again.

A. Create a syntax error on the last line (e.g. remove the "{" ).

B. Change the line ADD to SUB.

```
MOV R2, #44
SUB R3, R1, R2
BL printf
```

What happens if you do not compile the program after making a change?

# Assembly Language Addition Calculator with user input

1. Create the source file for the program

```
pi@raspberrypi:~/asm_calc $ nano adder2.s
```

2. Write the following code to the source file

```
 GNU nano 2.7.4                         File: adder2.s

.data
        string: .asciz "%d + %d = %d \n"
        prompt1: .asciz "Enter a number: "
        prompt2: .asciz "Enter another number: "
        num1: .word 0
        num2: .word 0
        scan: .asciz "%d"
.text
.global main

main:
        PUSH {ip, lr}
        LDR R0, =prompt1
        BL printf
        LDR R1, =num1
        LDR R0, =scan
        BL scanf
        LDR r0, =prompt2
        BL printf
        LDR R1, =num2
        LDR R0, =scan
        BL scanf
        LDR R6, =num1
        LDR R1, [R6]
        LDR R6, =num2
        LDR R2, [R6]
        ADD R3, R1, R2
        LDR R0, =string
        BL printf

        MOV r0, #0
        POP {ip, pc}
```

3. Press ctrl + o to write out the file
4. Press ctrl + x to return to the command line

## Assemble the source file

5. Assemble the source file (adder1.s) then compile the translated assembly to object code (adder1).

```
pi@raspberrypi:~/dev/asm_calc $ as adder2.s -o adder2.a
pi@raspberrypi:~/dev/asm_calc $ gcc adder2.a -o adder2
```

6. Run the program (object code)

```
pi@raspberrypi:~/asm_calc $ ./adder2
Enter a number: 96
Enter another number: 10
96 + 10 = 106
pi@raspberrypi:~/asm_calc $
```