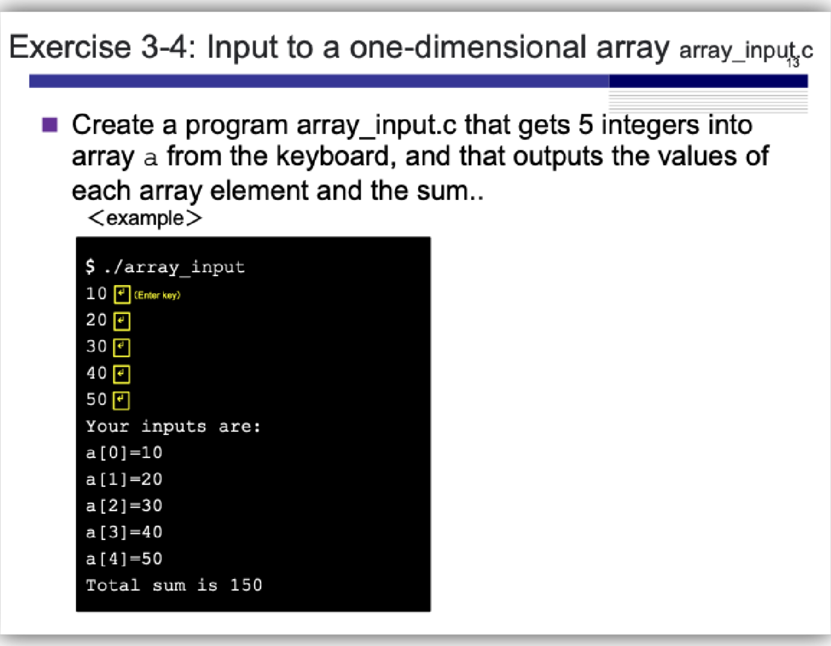
Exercise 3-4: Input to one-dimensional array (array_input.c)

Maximilian Fernaldy - C2TB1702



Exercise 3-4: Input to a one-dimensional array array_input.c 13

- Create a program array_input.c that gets 5 integers into array a from the keyboard, and that outputs the values of each array element and the sum..

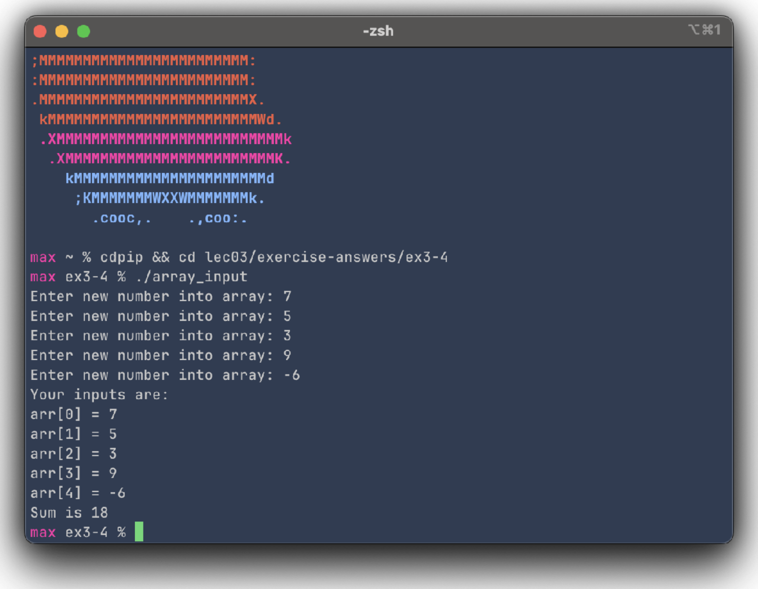
<example>

```
$ ./array_input
10 ↵ (Enter key)
20 ↵
30 ↵
40 ↵
50 ↵
Your inputs are:
a[0]=10
a[1]=20
a[2]=30
a[3]=40
a[4]=50
Total sum is 150
```

To repeat an action a certain number of times, we can use the `for` loop. First we initialize the iterator variable `i` and set it as 0, then set it to stop before the array length (because the highest index is equal to the array length minus one, due to the array being zero-indexed). Finally in the loop we ask for input from the user, and insert the input into the array with `arr[i] = new_entry;`.

```c
#include <stdio.h>

#define ARRAY_LENGTH 5

int main() {
    int arr[ARRAY_LENGTH], sum = 0, new_entry;
    for (int i = 0; i < ARRAY_LENGTH; i++) {
        printf("Enter new number into array: ");
        scanf("%d", &new_entry);
        arr[i] = new_entry;
        sum += arr[i];
    }

    printf("Your inputs are:\n");
    for (int i = 0; i < ARRAY_LENGTH; i++) {
        printf("arr[%d] = %d\n", i, arr[i]);
    }
    printf("Sum is %d\n", sum);

    return 0;
}
```

still inside the `for` brackets, we add the new entry to an accumulating variable `sum`. After the iterator variable `i` reaches 5, the condition for the `for` loop returns `False` and the loop stops after the fifth iteration. It then continues to execute the code block after the loop, printing the inputs with another for loop and finally printing out the sum of the input numbers. We can see that it also works with negative integers.



```
;MMMMMMMMMMMMMMMMMMMMMMMM:
:MMMMMMMMMMMMMMMMMMMMMMMM:
.MMMMMMMMMMMMMMMMMMMMMMMMX.
 kMMMMMMMMMMMMMMMMMMMMMMMMWd.
 .XMMMMMMMMMMMMMMMMMMMMMMMMMMk
  .XMMMMMMMMMMMMMMMMMMMMMMMMK.
    kMMMMMMMMMMMMMMMMMMMMMMd
     ;KMMMMMMMWXXWMMMMMMMk.
       .cooc,.    .,coo:.

max ~ % cdpip && cd lec03/exercise-answers/ex3-4
max ex3-4 % ./array_input
Enter new number into array: 7
Enter new number into array: 5
Enter new number into array: 3
Enter new number into array: 9
Enter new number into array: -6
Your inputs are:
arr[0] = 7
arr[1] = 5
arr[2] = 3
arr[3] = 9
arr[4] = -6
Sum is 18
max ex3-4 %
```