WARWICK

SCHOOL OF ENGINEERING

Module Code:
ES2C4

Module Title:
Computer Architecture and Systems

Learning Activity:
Lab. 3 C Programming

Learning Objectives:

- **Represent different types of data in binary and perform arithmetic operations on them**
- **Write C programs using bitwise operations**
- **Carry out arithmetic and logical operations on arrays**

Instructor:
Dr Sam Agbroko sam.agbroko@warwick.ac.uk

1. Introduction

In this lab session, you will carry out arithmetic and logical operations on variables. You will also practice carrying out computation in C programming language.

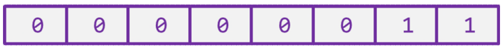
2. Guided tutorial - Bitwise operations

The task in this tutorial is to:

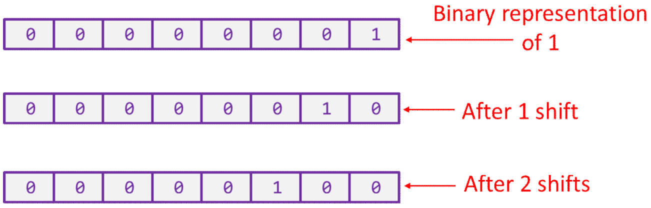
1. Write a C program to set the n-th bit of a number.
2. The program must ask the user to input a decimal number and index of the bit the user would like to set.
3. The program must output the new decimal number after setting the appropriate bit.

The sequence of instructions for this operation is:

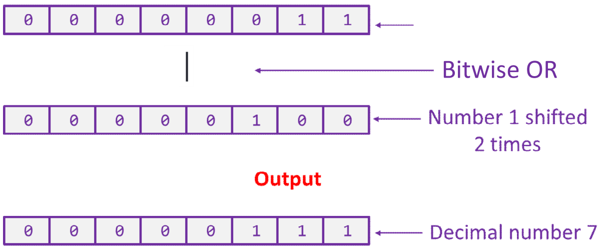
1. Take an input number from the user. If the user enters the number 3, this is represented as:



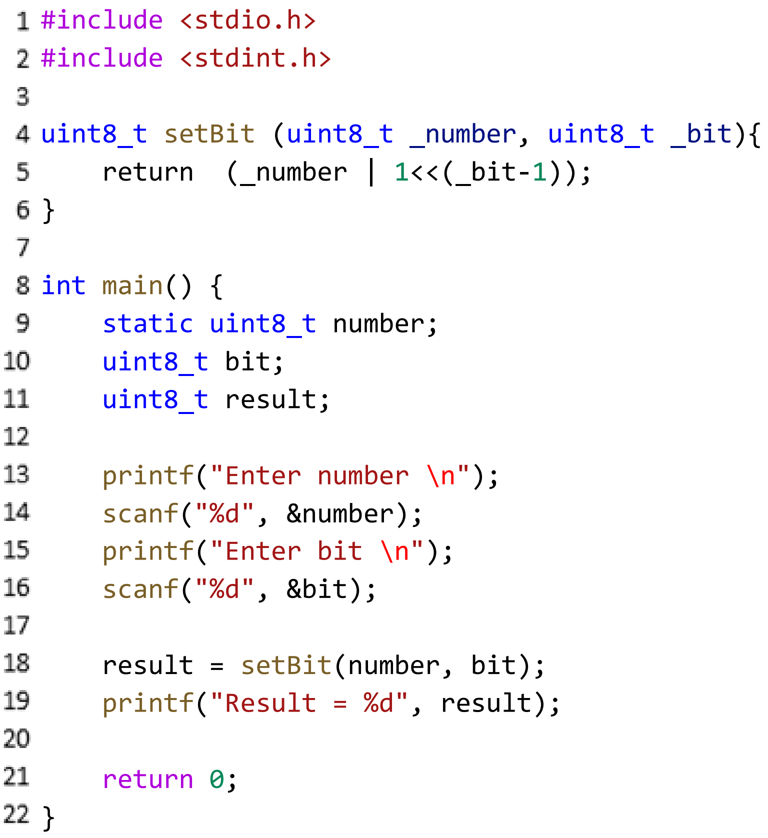
2. The user enters the bit to set, in this example, the user selects the 3rd bit.
3. To set the third bit, left shift the decimal number 1 by ‘n-1’ times.



4. Carry out a logical OR between the input number and the shifted number to obtain the result.



The program for this operation is shown below:



```
 1 #include <stdio.h>
 2 #include <stdint.h>
 3
 4 uint8_t setBit (uint8_t _number, uint8_t _bit){
 5     return  (_number | 1<<(_bit-1));
 6 }
 7
 8 int main() {
 9     static uint8_t number;
10     uint8_t bit;
11     uint8_t result;
12
13     printf("Enter number \n");
14     scanf("%d", &number);
15     printf("Enter bit \n");
16     scanf("%d", &bit);
17
18     result = setBit(number, bit);
19     printf("Result = %d", result);
20
21     return 0;
22 }
```

There are some new concepts to note:

1. Line 2: the **stdint.h** header file was added therefore the program can use unsigned 8-bit integer types (**uint8_t**). Since the value entered by the user will not exceed 8 bits, using this type enables efficient use of memory resources.
2. Line 9: The **static** keyword is used to ensure that the number variable is preserved for the duration of the program.

3. Exercises

(i) Write a C program that takes an integer input from a user and checks whether the Least Significant Bit (LSB) of the given number is set (1) or not (0).

(ii) Write a C program to input any number from user and check whether the nth bit of the given number is set (1) or not (0).

(iii) Write a program that generates a random number between 0 and 10. The generated number should be different each time the program is executed.
Hint: **rand()**, **srand()**, **time.h**, **stdlib.h**, **time()**, **%**

(iv) Write a C program to sort an array in ascending and descending order.