Tutorial 3: Bit Operations and Structs

120.3: Programming III, C

1 Bit Operations

You are given an integer `int n = 1288243249`.

1. Take the function `printBits` from Lecture 3 and alter it so that it prints the first 16 bits of the integer passed to the function followed by a space and the second 16 bits. Print out `n` using this function.

2. Assign the first and second 16 bits of `n` to two 16-bit unsigned integers `first` and `second`. Use a hexadecimal mask to achieve this.

3. Create a function `printBits16` which takes as a parameter 16-bit unsigned integer and prints its bitwise representation. Write this function so that the mask is shifted instead of the integer parameter. Use it to print out the bits of `first` and `second` with a space between them so that it produces the same output as your earlier call to `printBits`.

4. Print out `first` and `second` using the function `printBits`.

5. Create a signed 16-bit integer `signedFirst` and assign `first` to it. Print out the decimal values of both of these integers.

6. Shift `first` left by 1 bit and assign it to `signedFirst`. Print out the decimal values both of these integers.

7. Create a function `printSBits16` which takes as a parameter 16-bit signed integer and prints out the bits. Print out the bits of `first` using `printBits16` and the bits of `signedFirst` using `printSBits16`.

2 Structs

1. Create a `struct` representing a person which contains their name (`char name[50]`) and their age (`int age`)

2. Create a function `getPerson(void)` which reads in the name and age of a person and returns a pointer to a struct (of the type defined above) containing these details.

3. Create a function `printPeople` which takes two parameters: an array of pointers to structs (of the type defined above) and an integer which indicates the number of people to print out. Print out the name and age of each person in the array.

4. In the main function read in a small number of people using the `getPerson(void)` function, store them in an array, and print them out using the `printPeople` function. Make sure any memory used is freed.