

1806ICT Programming Fundamentals

Bitwise Operators, Enumerations, Macros, Recursion

1. The constant `CHAR_BIT` is defined in the header file `limits.h` to represent the number of bits in a `char` or byte. Now, write two functions that counts the number of bits in a `char` and an `int`.

Hint: The expression `~0` (complement of 0) will produce a number that contains bits that are all 1's.

2. Left shifting an unsigned integer by 1 bit is the same as multiplying that integer by the value 2. Write a function that takes in two unsigned integer parameters `number` and `power`, and computes the value of $(\text{number} \times 2^{\text{power}})$.

Sample run:

Input	Output
3 4	48
5 2	20

3. Write a recursive function that computes the sum of the first `n` positive integers. Your program will read in the integer value for `n`, and call the recursive function to compute the sum of $1+2+3+\dots+n$.
4. The greatest common divisor of integers `x` and `y` is the largest integer that evenly divides both `x` and `y`. Write a recursive function `gcd` that returns the greatest common divisor of `x` and `y`. The `gcd` of `x` and `y` is defined recursively as follows:

If `y` is equal to 0, then `gcd(x, y)` is `x`;
otherwise `gcd(x, y)` is `gcd(y, x % y)` where `%` is the remainder operator.

5. Write a recursive program to find the largest number in an array of integer numbers. The prototype for the recursive function is `largest(int array[], int startIndex, int endIndex)`, where `startIndex` is 0 and `endIndex` is the index of the last element in the array.

Sample run:

Input	Output
1 2 3 4 5 6	6
1 4 2 65 3 23	65

6. Write a recursive function `stringReverse()` that takes a character array as an argument and prints it back to front.

Sample run:

Input	Output
hello	olleh
goodbye	eybdoog