Roman Millem

Assignment 1

September 16, 2018

Problem 1:

The purpose of this problem is to utilize recursion to represent the number of 1's in the binary representation of a decimal number.

My program asks for a user to input a decimal number, which then is sent to a recursive function BinOnes(). The method of conversion from decimal to binary is recursive in itself. The conversion is finished when the number N divided by 2 returns 0 with a remainder of 1. So I used that as the base case. The number N is divided by 2. The quotient (N/2) is used in the recursive call. Each time this calculation is made, if there is a remainder (N%2 == 1), the recursive call will be made and 1 will be added to the result. This results in a natural counter that is incremented each time there is a remainder, a 1, in the binary representation.

The input must be an int and the program will simply return the number of 1's in the binary representation.

```
Enter a number: 5
2

Process returned 0 (0x0)   execution time : 7.572 s
Press any key to continue.
```

```
Enter a number: 11
3

Process returned 0 (0x0)   execution time : 1.987 s
Press any key to continue.
```

```
Enter a number: 15
4

Process returned 0 (0x0)   execution time : 1.877 s
Press any key to continue.
```

```
Enter a number: 125
6

Process returned 0 (0x0)   execution time : 7.801 s
Press any key to continue.
```

Problem 2:

Two routines declarations are given that must be used. The first must call the second, printing all permutations of a string.

A string is received from the user which is then passed by reference to the first routine that accepts only a string reference. Using a recursive function, all positions for each character are found without using the same on twice. For example, if given a string of 3 characters, character at index 0 will be swapped with itself, index [1], and index [2]. The resulting string from each of these swaps are then passed through the function again, this time swapping characters starting at index [1]. So, the character at index [1] will be swapped with itself and with index [2]. The base case is reached when the starting swap index is equal to the string length.

```
Enter a string: abc
abc
acb
bac
bca
cab
cba

Process returned 0 (0x0)   execution time : 1.487 s
Press any key to continue.
```

```
Enter a string: abcd
abcd
abdc
acbd
acdb
adbc
adcb
bacd
badc
bcad
bcda
bdac
bdca
cabd
cadb
cbad
cbda
cdab
cdba
dabc
dacb
dbac
dbca
dcab
dcba

Process returned 0 (0x0)   execution time : 3.621 s
Press any key to continue.
```

For a string of length L, there are L! permutations (Ex: L = 4, N=4*3*2*1=24 permutations)

Problem 3: