

CECS 302-01 Assignment #1

Nicholas Gittings

September 18, 2018

To solve the first problem, recognize that binary is base 2. Now create a function called binary ones that accepts an integer N as a parameter. Next, create an if statement that if N is equal to 0 it returns 0 and create a else if statement that if N is equal to 1 return 1. Lastly create the else statement which will use recursion, return the remainder of N divided by 2 + binary ones (n/2). This will result in the function returning how many binary ones are in the integer N. The number of binary ones of N is displayed to the user.

```
Homework Client V1.0
1. Get number of binary ones from integer
2. Get permutations for string
3. Compute nearest neighbour using vector of points
4. Exit
1
Input an integer:
4
1
Homework Client V1.0
1. Get number of binary ones from integer
2. Get permutations for string
3. Compute nearest neighbour using vector of points
4. Exit
1
Input an integer:
7
3
```

Screenshot of binary ones

To solve the second problem, create two routines called permute, one having a string input parameter and the second one having the string input, integer low, and integer high. The first routine acts as a driver which takes the string and sets high to the length of the string, low as 0, and calls the next routine. The second routine takes the parameters from the previous routine and has an if else statement for if low is equal to high then the string is one character and will be returned, else it will find the permutations. To find the permutations of the string, use recursion and the swap function. This will

set one of the characters of the string as constant and proceed to find the permutations and keep characters constant till it finishes then moves on to the next character and repeats. Each permutation is displayed to the user.

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

Screenshot of permutations of string "abc"

To solve the third problem, create a function called `closest` which accepts a `pair<double, double>`, `vector<pair<double, double>>`, and integer. The function takes a set of house coordinates, a vector of coordinates nearby and will find the closest coordinates to the house. First, create two functions called `d_eulc` and `d_man`, which are functions that accept 2 x values and 2 y values to calculate distances in Euclidean and Manhattan which will be using an integer of 1 for Euclidean and 0 for Manhattan. Next create a for loop inside the `closest` function which will go through the vector of points and compute distance from the house coordinates given by the user. These distances are put into a new vector of doubles which then is used to find the lowest value using the `min` element function. Lastly, set up another for loop to calculate the distances again and this time set up an if statement to find the distance that is equal to the min distance found before, then display it to the user.

```
Homework Client V1.0
1. Get number of binary ones from integer
2. Get permutations for string
3. Compute nearest neighbour using vector of points
4. Exit
3
Enter x coordinate of house
100
Enter y coordinate of house
100
Enter a x coordinate:
50
Enter a y coordinate:
50
Done? (Y/N)
n
Enter a x coordinate:
40
Enter a y coordinate:
40
Done? (Y/N)
n
Enter a x coordinate:
80
Enter a y coordinate:
80
Done? (Y/N)
y
Enter 1 for Euclidean or 0 for Manhattan
1
Nearest neighbour is: (80,80) at 28.2843 euclidean distance
```

Screenshot of finding nearest neighbor