"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 4.1): octal to decimal

Pseudocode:

Input octal

decimal = 0

exponent = len(octal) - 1

For digit In octal:

If digit > 7 Then Output error message & Quit

decimal += digit \* 8^exponent

exponent -= 1

Output decimal

"""

octal = input("Enter an octal number: ")

decimal = 0

exponent = len(octal) - 1

for digit in octal:

if int(digit) > 7:

print("Invalid octal number...")

exit()

decimal += int(digit) \* 8 \*\* exponent

exponent -= 1

print("")

print("The decimal value is: ", decimal)

"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 4.2): decimal to octal

Pseudocode:

Input decimal (integer)

octal = ""

if decimal == 0

print(0)

else

while decimal > 0

remainder = decimal % 8

decimal = decimal // 8

octal = str(remainder) + octal

Output octal

"""

decimal = int(input("Enter an decimal integer: "))

print("")

octal = ""

if decimal == 0:

print(0)

else:

while decimal > 0:

remainder = decimal % 8

decimal = decimal // 8

octal = str(remainder) + octal

print("The octal value is: ", octal)

"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 5.1): bit shift left

Pseudocode:

Input bitString

leftShift = bitString[1:length of bitString] + bitString[0]

Output leftShift

"""

bitString = input("Enter a binary string:")

leftShift = bitString[1:len(bitString)] + bitString[0]

print("After left shift: ", leftShift)

"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 5.2): bit shift right

Pseudocode:

Input bitString

rightShift = bitString[last character] + bitString[0:length of bitString - 2]

Output rightShift

"""

bitString = input("Enter a binary string:")

rightShift = bitString[-1] + bitString[0:len(bitString) - 1]

print("After right shift: ", rightShift)

"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 6):

encrypt a message by converting the characters in it to their ASCII value,

then converting the ASCII value to a binary value,

then shifting the digits in that binary value one place to the left

Pseudocode:

Input message

encryption = ""

For index = 0 To length of message

asciiDec = ASCII value of message[index]

asciiBin = ""

While asciiDec > 0

remainder = asciiDec % 2

asciiDec = asciiDec // 2

asciiBin = String(remainder) + asciiBin

asciBin = asciiBin[1:length of asciiBin] + asciiBin[0]

encryption = encryption asciiBin + " "

Output encryption

"""

message = input("Enter a message to be encoded: ")

encryption = ""

for index in range(0, len(message)):

asciiDec = ord(message[index])

asciiBin = ""

while asciiDec > 0:

remainder = asciiDec % 2

asciiDec = asciiDec // 2

asciiBin = str(remainder) + asciiBin

asciiBin = asciiBin[1:len(asciiBin)] + asciiBin[0]

encryption += asciiBin + " "

print("The encoded message is: ", encryption)

"""

Student: Robin G. Blaine

Date: October 18, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 4, Project 7):

write a program to decrypt a message encrypted by Project 6

Pseudocode:

Input message

decryption = ""

listOfCharacters = message.split

For binary In listOfCharacters

binary = binary[last character] + binary[0 to next-to-last character]

asciiCode = 0

exponent = len(binary) - 1

For digit In binary

asciiCode = asciiCode + digit \* 2^exponent

exponent = exponent - 1

descryption = descryption + character(asciiCode)

Output decryption

"""

message = input("Enter a message to be decoded: ")

decryption = ""

listOfCharacters = message.split()

for binary in listOfCharacters:

binary = binary[-1] + binary[0:len(binary) - 1]

asciiCode = 0

exponent = len(binary) - 1

for digit in binary:

asciiCode += int(digit) \* 2 \*\* exponent

exponent -= 1

decryption += chr(asciiCode)

print("The decoded message is: ", decryption)