**CMPS 5243 Project 1**

Copy/paste the code below into an IDE of your choice (it was tested with Replit).

def change(x):

if x != 0:

return str(change(x // 2)) + str(x % 2)

else:

return ""

num = int(input("Enter a positive integer: "))

result = str(change(num))

print(result)

Modified Code with comments:

def change(x):

"""

This function used recursion to convert

an integer input into its binary equivalent

and output it as a string to the console.

"""

if x != 0: # check that the input is >0

return str(change(x // 2)) + str(x % 2)

# ‘*x // 2’* divides variable by 2 & returns integer value without rounding up

# ’*x % 2’* divides the variable by 2 & returns remainder (either 1 or 0)

# *str* converts int type into a string type & concatenates the modulo portion

# to a growing string that represents the binary equivalent of the original variable x

else:

return ""

""" loop to check input for valid type and sign"""

while True:

try:

num = int(input("Enter a positive integer: "))

if num<0 :

raise ValueError("Please enter a positive integer")

break

except ValueError as e:

print(f"Error: {e}. Try again")

result = str(change(num))

print(result)

1. Run the program several times and analyze the code for understanding. Type comments above the function explaining its purpose. The comments can be typed in this document or in the IDE and then copy/pasted back into this document. Ensure that the comments are readable (spelling, grammar, appropriate word wrap). (20 points)
2. Complete the trace of the code below. (20 points)

change(19) = str(change(9)) + "1" 🡪 generated string: 1

change(9) = str(change(4)) + “1” 🡪 generated string: 11

change(4) = str(change(2)) + “0” 🡪 generated string: 011

change(2) = str(change(1) + “0” 🡪 generated string: 0011

change(1) = str(change(0) + “1” 🡪 generated string: 10011

base case reached (x=0)

Final result: change(19) = “10011”

1. What is the complexity (Big-O) of the recursive function? Explain how you determined your answer. (60 points)

Complexity = O(n)

Original definition

<https://youtu.be/ASGgn8bNQuA?si=DEot40mhx6D7SqWV>

subsequent usage:

<https://youtu.be/tdpjxFbFJ9M?si=h8LMUaXq7WSjVO5B>

most recent incarnation:

<https://youtu.be/xAHQ4akjlmo?si=Fp8SWCdg5Fp2yB55>