



**DALHOUSIE
UNIVERSITY**

Faculty of Computer Science

CSCI 6704 – Advanced Topics in Networks

Name: Arka Ghosh

Banner ID: B00911033

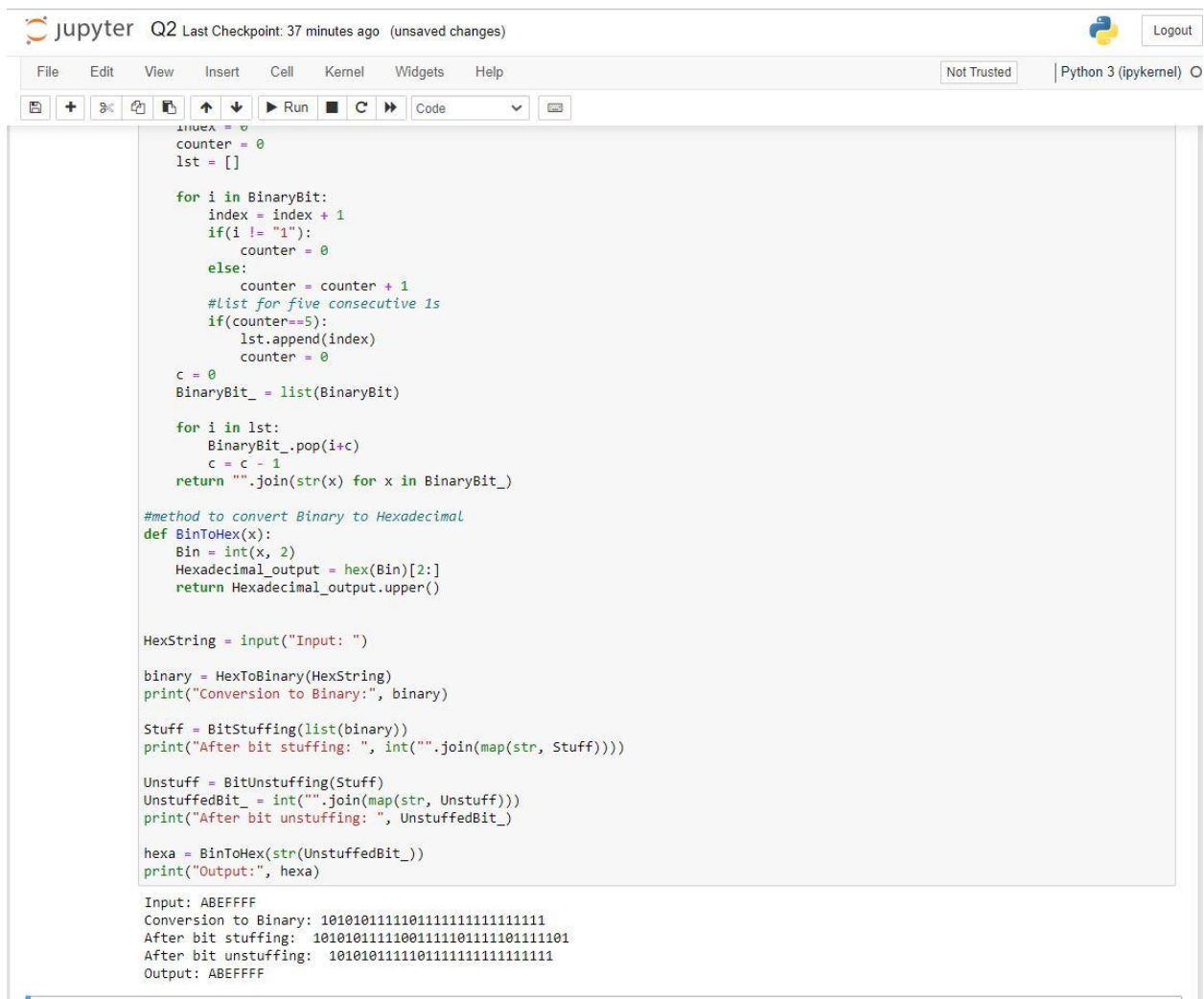
Assignment: 01 (Question 2)

Sample Input and Output for Question 2

Three hexadecimal strings have been taken to test the program for this question which are ABEFFFF, FFFFFFFF and ABCDFF. The resulting outputs with the screenshot have been added below:

Test Case 1:

Input: ABEFFFF
Conversion to Binary: 1010101111101111111111111111
After bit stuffing: 101010111110011111101111101111101
After bit unstuffing: 1010101111101111111111111111
Output: ABEFFFF



```
jupyter Q2 Last Checkpoint: 37 minutes ago (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
index = 0
counter = 0
lst = []

for i in BinaryBit:
    index = index + 1
    if(i != "1"):
        counter = 0
    else:
        counter = counter + 1
    #List for five consecutive 1s
    if(counter==5):
        lst.append(index)
        counter = 0

c = 0
BinaryBit_ = list(BinaryBit)

for i in lst:
    BinaryBit_.pop(i+c)
    c = c + 1
return "".join(str(x) for x in BinaryBit_)

#method to convert Binary to Hexadecimal
def BinToHex(x):
    Bin = int(x, 2)
    Hexadecimal_output = hex(Bin)[2:]
    return Hexadecimal_output.upper()

HexString = input("Input: ")

binary = HexToBinary(HexString)
print("Conversion to Binary:", binary)

Stuff = BitStuffing(list(binary))
print("After bit stuffing: ", int("".join(map(str, Stuff))))

Unstuff = BitUnstuffing(Stuff)
UnstuffedBit_ = int("".join(map(str, Unstuff)))
print("After bit unstuffing: ", UnstuffedBit_)

hexa = BinToHex(str(UnstuffedBit_))
print("Output:", hexa)

Input: ABEFFFF
Conversion to Binary: 1010101111101111111111111111
After bit stuffing: 101010111110011111101111101111101
After bit unstuffing: 1010101111101111111111111111
Output: ABEFFFF
```

Test Case 2:

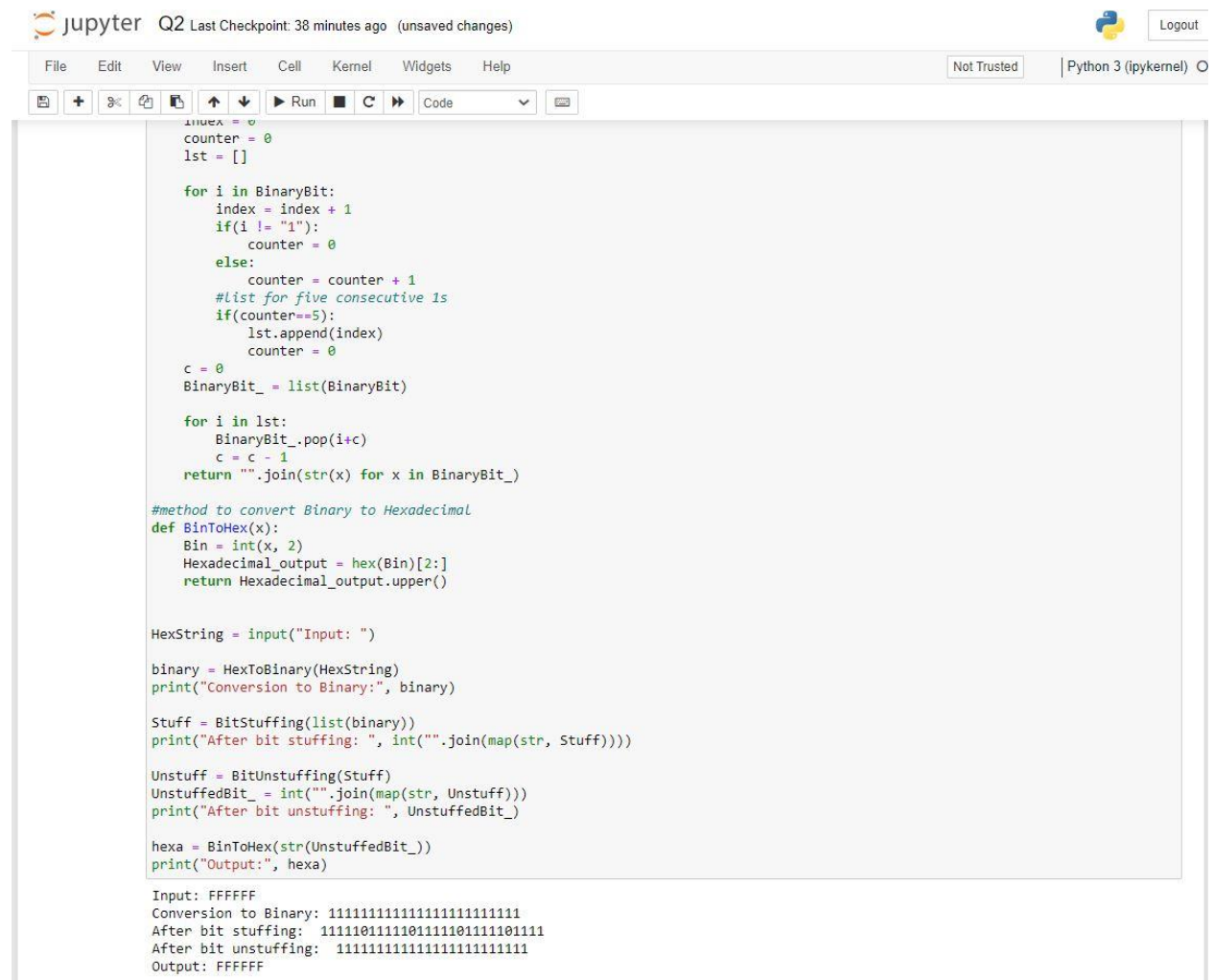
Input: FFFFFFFF

Conversion to Binary: 111111111111111111111111

After bit stuffing: 1111101111101111101111101111

After bit unstuffing: 111111111111111111111111

Output: FFFFFFFF



The image shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook is titled "Q2" and shows the last checkpoint was 38 minutes ago. The code in the notebook is as follows:

```
index = 0
counter = 0
lst = []

for i in BinaryBit:
    index = index + 1
    if(i != "1"):
        counter = 0
    else:
        counter = counter + 1
    #List for five consecutive 1s
    if(counter==5):
        lst.append(index)
        counter = 0

c = 0
BinaryBit_ = list(BinaryBit)

for i in lst:
    BinaryBit_.pop(i+c)
    c = c - 1
return "".join(str(x) for x in BinaryBit_)

#method to convert Binary to Hexadecimal
def BinToHex(x):
    Bin = int(x, 2)
    Hexadecimal_output = hex(Bin)[2:]
    return Hexadecimal_output.upper()

HexString = input("Input: ")

binary = HexToBinary(HexString)
print("Conversion to Binary:", binary)

Stuff = BitStuffing(list(binary))
print("After bit stuffing: ", int("".join(map(str, Stuff))))

Unstuff = BitUnstuffing(Stuff)
UnstuffedBit_ = int("".join(map(str, Unstuff)))
print("After bit unstuffing: ", UnstuffedBit_)

hexa = BinToHex(str(UnstuffedBit_))
print("Output:", hexa)

Input: FFFFFFFF
Conversion to Binary: 111111111111111111111111
After bit stuffing: 1111101111101111101111101111
After bit unstuffing: 111111111111111111111111
Output: FFFFFFFF
```

Test Case 3:

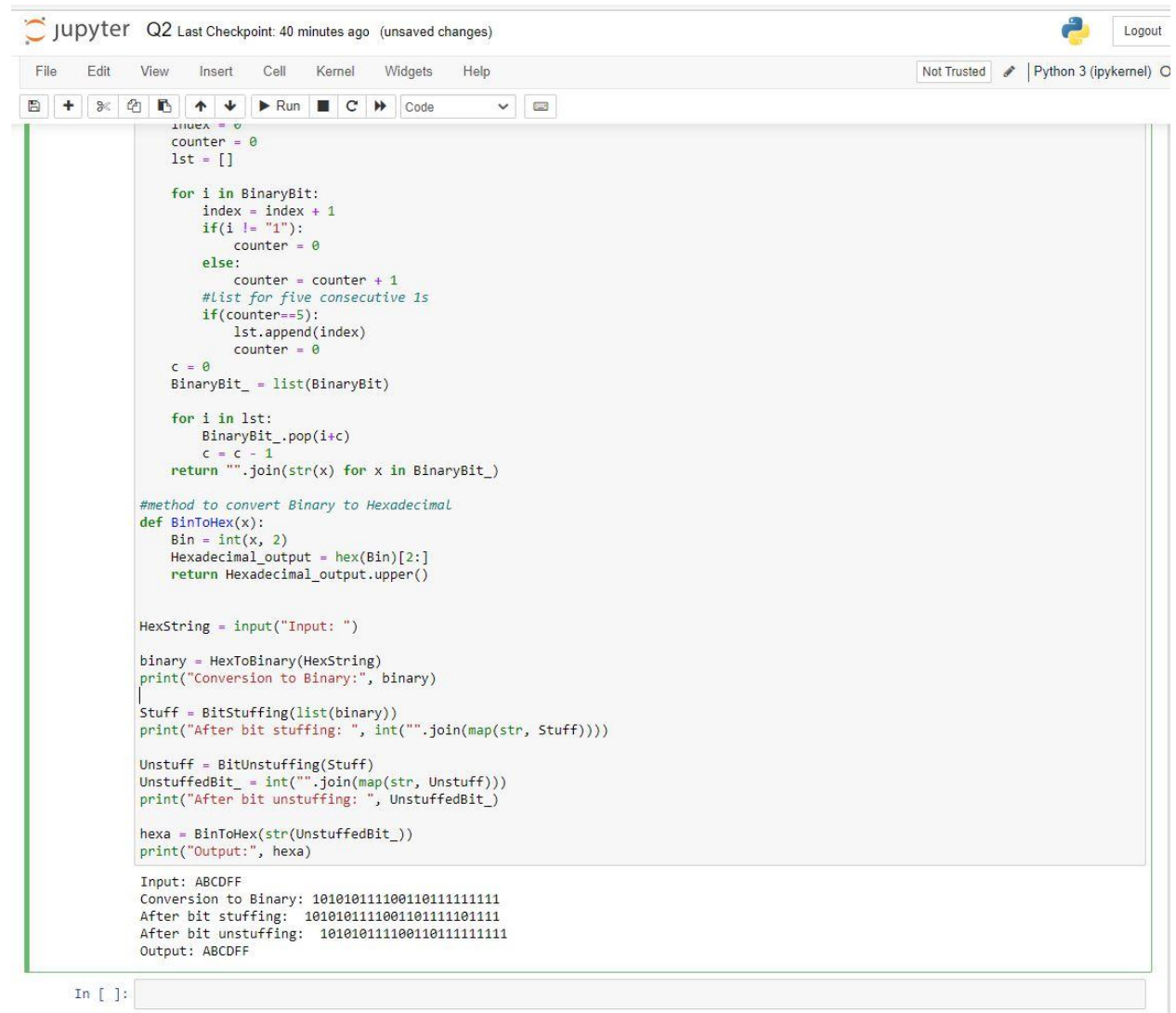
Input: ABCDFF

Conversion to Binary: 101010111100110111111111

After bit stuffing: 10101011110011011111101111

After bit unstuffing: 101010111100110111111111

Output: ABCDFF



The image shows a Jupyter Notebook interface with a Python script. The script defines functions for converting hexadecimal to binary, performing bit stuffing, and unstuffing. It then takes the input 'ABCDFF', converts it to binary, performs bit stuffing, unstuffing, and finally converts the result back to hexadecimal, which remains 'ABCDFF'.

```
index = 0
counter = 0
lst = []

for i in BinaryBit:
    index = index + 1
    if(i != "1"):
        counter = 0
    else:
        counter = counter + 1
    #List for five consecutive 1s
    if(counter==5):
        lst.append(index)
        counter = 0

c = 0
BinaryBit_ = list(BinaryBit)

for i in lst:
    BinaryBit_.pop(i+c)
    c = c + 1
return "".join(str(x) for x in BinaryBit_)

#method to convert Binary to Hexadecimal
def BinToHex(x):
    Bin = int(x, 2)
    Hexadecimal_output = hex(Bin)[2:]
    return Hexadecimal_output.upper()

HexString = input("Input: ")

binary = HexToBinary(HexString)
print("Conversion to Binary:", binary)
|
Stuff = BitStuffing(list(binary))
print("After bit stuffing: ", int("".join(map(str, Stuff))))

Unstuff = BitUnstuffing(Stuff)
UnstuffedBit_ = int("".join(map(str, Unstuff)))
print("After bit unstuffing: ", UnstuffedBit_)

hexa = BinToHex(str(UnstuffedBit_))
print("Output:", hexa)

Input: ABCDFF
Conversion to Binary: 101010111100110111111111
After bit stuffing: 10101011110011011111101111
After bit unstuffing: 101010111100110111111111
Output: ABCDFF
```

In []: