**P4**

a.

01011100 + 01100101 = 11000001

Its one's complement : 00111110

b.

11011010 + 01100101 = 01000000

Its one's complement : 10111111

c.

The first byte: 01010100

The second byte: 01101101

Program:

|  |  |
| --- | --- |
|  |  |
|  | NumZero = 2 \*\* 16 |
|  | NumOne = 0B0110011001100000 |
|  | NumTwo = 0B0101010101010101 |
|  | NumThree = 0B1000111100001100 |
|  | UDPList = [NumOne, NumTwo, NumThree] |
|  | print("original udp data package") |
|  | print(UDPList) |
|  |  |
|  | def Complement(num): |
|  | return 0xffff - num & 0xffffffff |
|  |  |
|  | def CalcSum(num01, num02): |
|  | num03 = num01 + num02 |
|  | num04 = num03 % NumZero |
|  | if num04 < num01 or num04 < num02: |
|  | num04 = num04 + 1 |
|  | return num04 |
|  |  |
|  |  |
|  | def CheckSum(ParaList): |
|  | check = ParaList[0] |
|  | for i in range(len(ParaList) - 1): |
|  | check = CalcSum(check, ParaList[i + 1]) |
|  | UDPList.append(Complement(check)) |
|  | return |
|  |  |
|  |  |
|  | def Distinguish(ParaList): |
|  | result = 0 |
|  | check = ParaList[0] |
|  | for i in range(len(ParaList) - 1): |
|  | check = CalcSum(check, ParaList[i + 1]) |
|  | result += (check == NumZero - 1) |
|  | return result |
|  |  |
|  |  |
|  | CheckSum(UDPList) |
|  | print("Add Check Number") |
|  | print(UDPList) |
|  | print("Start Checking") |
|  | if Distinguish(UDPList): |
|  | print("Success") |
|  | else: |
|  | print("Fail") |
|  |  |
|  | plot.scatter([1, 2, 3], [UDPList[:3]], color="blue", marker="^") |
|  | plot.scatter(4, UDPList[3], color="red", marker="o") |
|  | plot.show() |