DCCN LAB 2

Name: AMAN KUMAR GUPTA

Roll no: BTech/25013/18

Branch: CSE

Program 1

Aim : Write a program to determine if the IP address is in Class A, B, C, D, or E

Code:

```
def find_class(ip):
    ip = list(map(int,ip.split(".")))
    if ip[0]>=0 and ip[0]<128:
        return "A"
    elif ip[0]>=128 and ip[0]<192 :
        return "B"
    elif ip[0]>=192 and ip[0]<224:
        return "C"
    elif ip[0]>=224 and ip[0]<240 :
        return "D"
    elif ip[0]>=240 and ip[0]<256 :
        return "E"

ip = input("Enter ip Address : ")
print()
print(f"Entered IP address is in Class : {find_class(ip)} \n")
```

Input and Output :

```
PS C:\Users\LENOVO\Desktop\BTech_6_Lab> & C:/Users/LENOVO/anaconda3/python.exe c
:/Users/LENOVO/Desktop/BTech_6_Lab/dccn/ip_class.py
Enter ip Address : 220.100.8.12

Entered IP address is in Class : C

PS C:\Users\LENOVO\Desktop\BTech_6_Lab> & C:/Users/LENOVO/anaconda3/python.exe c
:/Users/LENOVO/Desktop/BTech_6_Lab/dccn/ip_class.py
Enter ip Address : 128.1.2.128

Entered IP address is in Class : B
```

Program 2

Aim : Write a program to translate dotted decimal IP address into 32 bit address.

Code:

```
def convert_ip_dec_bin(ip):
    ip_dec = list(map(int,ip.split(".")))
    return '{:08b}.{:08b}.{:08b}.{:08b}'.format(*ip_dec)

ip = input("\nEnter Decimal ip Address : ")
print()
print(f"32-bit conversion of ip {ip} : \n{convert_ip_dec_bin(ip)} \n")
```

Input and Output :

```
PS C:\Users\LENOVO\Desktop\BTech_6_Lab> & C:/Users/LENOVO/anaconda3/python.exe c
:/Users/LENOVO/Desktop/BTech_6_Lab/dccn/dec_bin_ip.py

Enter Decimal ip Address : 128.1.2.128

32-bit conversion of ip 128.1.2.128 :
10000000.00000001.00000010.10000000

PS C:\Users\LENOVO\Desktop\BTech_6_Lab> & C:/Users/LENOVO/anaconda3/python.exe c
:/Users/LENOVO/Desktop/BTech_6_Lab/dccn/dec_bin_ip.py

Enter Decimal ip Address : 220.100.8.12

32-bit conversion of ip 220.100.8.12 :
11011100.01100100.00001000.00001100
```

Program 3

Aim : Write a program to perform bit stuffing and de-stuffing.

Code:

```python
def stuffing(bit_str):
    bits = list(bit_str)
    stuffing_count = 0
    for i,bit in enumerate(bits):
        if bit == '1':
            stuffing_count+= 1
        if bit == '0':
            stuffing_count = 0
        if stuffing_count == 5:
            stuffing_count = 0
            bits.insert(i+1,"0")
    return ''.join(bits)

def destuffing(bit_str):
    bits = list(bit_str)
    destuffing_count = 0
    for i,bit in enumerate(bits):
        if destuffing_count == 5 and bit == "0":
            destuffing_count = 0
            del bits[i]
        if bit == '1':
            destuffing_count+= 1
        if bit == '0':
            destuffing_count = 0
    return ''.join(bits)

bit_str = input("\nEnter the Bit String : ")

print(f"\nOriginal String : {bit_str}\n")
print(f"Stuffed String : {stuffing(bit_str)}\n")
print(f"Destuffed String : {destuffing(bit_str)}\n")
```

Input and Output :

```
PS C:\Users\LENOVO\Desktop\BTech_6_Lab> & C:/Users/LENOVO/anaconda3/python.exe c:
/Users/LENOVO/Desktop/BTech_6_Lab/dccn/bit_stuffing_destuffing.py

Enter the Bit String : 1110010111001111111

Original String : 1110010111001111111

Stuffed String : 11100101110011111011

Destuffed String : 1110010111001111111
```