

Ques 1. Implement the error correcting code.

Ans:-

CSC/20/50 Bharat Sharma Univ_Roll_No:- 20059570040

```
def hamming_correct(code):  
    # Calculate the number of parity bits.  
    n = len(code)  
    r = 0  
    while 2**r <= n:  
        r += 1  
    # Generate the syndrome.  
    syndrome = 0  
    for i in range(r):  
        pos = 2**i - 1  
        bit = 0  
        for j in range(pos, n, 2*pos + 2):  
            for k in range(pos + 1):  
                if j + k >= n:  
                    break  
                if (k != pos):  
                    bit = bit ^ int(code[j + k])  
            syndrome += bit * (2**i)  
    # If the syndrome is non-zero, correct the error.  
    if syndrome > 0:  
        # Flip the bit at the position indicated by the syndrome.  
        pos = syndrome - 1  
        if pos < n:  
            code = code[:pos] + str(int(not int(code[pos]))) + code[pos+1:]  
    return code  
code = input("Enter code : ")  
# Correct the error in the code.  
corrected_code = hamming_correct(code)  
# Print the original code and the corrected code.  
print("Original code: ", code)  
print("Corrected code: ", corrected_code)
```

OUTPUT:-

```
IS_Ques1 - Jupyter Notebook
localhost:8888/notebooks/IS_Ques1.ipynb

jupyter IS_Ques1 Last Checkpoint: 10 minutes ago (autosaved)
Python 3 (ipykernel)

In [1]: # CSC/20/50 Bharat Sharma Univ_Roll_No:- 20059570040
def hamming_correct(code):
    # Calculate the number of parity bits.
    n = len(code)
    r = 0
    while 2**r <= n:
        r += 1

    # Generate the syndrome.
    syndrome = 0
    for i in range(r):
        pos = 2**i - 1
        bit = 0
        for j in range(pos, n, 2**i + 1):
            for k in range(pos + 1, n):
                if j + k >= n:
                    break
                if (k != pos):
                    bit = bit ^ int(code[j + k])
            syndrome += bit * (2**i)

    # If the syndrome is non-zero, correct the error.
    if syndrome > 0:
        # Flip the bit at the position indicated by the syndrome.
        pos = syndrome - 1
        if pos < n:
            code = code[:pos] + str(int(not int(code[pos]))) + code[pos+1:]

    return code
```

```
IS_Ques1 - Jupyter Notebook
localhost:8888/notebooks/IS_Ques1.ipynb

jupyter IS_Ques1 Last Checkpoint: 10 minutes ago (autosaved)
Python 3 (ipykernel)

syndrome += bit * (2**i)

# If the syndrome is non-zero, correct the error.
if syndrome > 0:
    # Flip the bit at the position indicated by the syndrome.
    pos = syndrome - 1
    if pos < n:
        code = code[:pos] + str(int(not int(code[pos]))) + code[pos+1:]

    return code

In [2]: code = input("Enter code : ")

# Correct the error in the code.
corrected_code = hamming_correct(code)

# Print the original code and the corrected code.
print("Original code: ", code)
print("Corrected code: ", corrected_code)

Enter code : 10011001
Original code: 10011001
Corrected code: 10011000
```