Name: Neeraj Varma

Roll. No: 8

Batch: A1

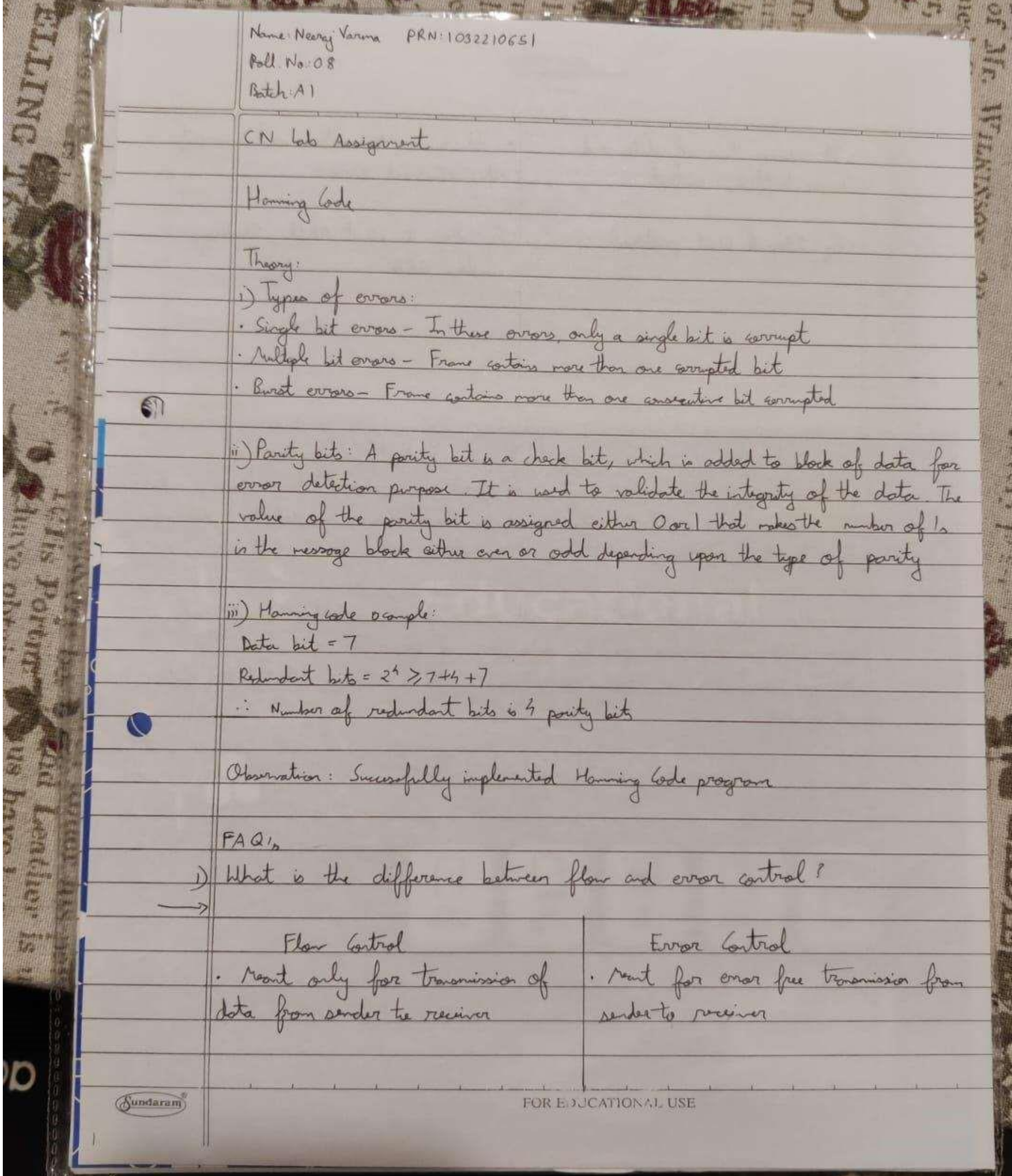
PRN: 1032210651

Title: Error detection and correction

Aim: To write a program for error detection and correction using Hamming Code

Objectives:

1. To encode and decode original data bits with the help of parity bits
2. To demonstrate use of error control protocols



Name: Neeraj Varma PRN: 1032210651
Roll. No.: 08
Batch: A1

CN lab Assignment

Hamming Code

Theory:

i) Types of errors:

- Single bit errors - In these errors, only a single bit is corrupt
- Multiple bit errors - Frame contains more than one corrupted bit
- Burst errors - Frame contains more than one consecutive bit corrupted

ii) Parity bits: A parity bit is a check bit, which is added to block of data for error detection purpose. It is used to validate the integrity of the data. The value of the parity bit is assigned either 0 or 1 that makes the number of 1s in the message block either even or odd depending upon the type of parity

iii) Hamming code example:

Data bit = 7

Redundant bits = $2^4 \geqslant 7+4+7$

$\therefore$ Number of redundant bits is 4 parity bit

Observation: Successfully implemented Hamming Code program

FAQ's

1) What is the difference between flow and error control?

→

Flow Control	Error Control
· Meant only for transmission of data from sender to receiver	· Meant for error free transmission from sender to receiver

Sundaram

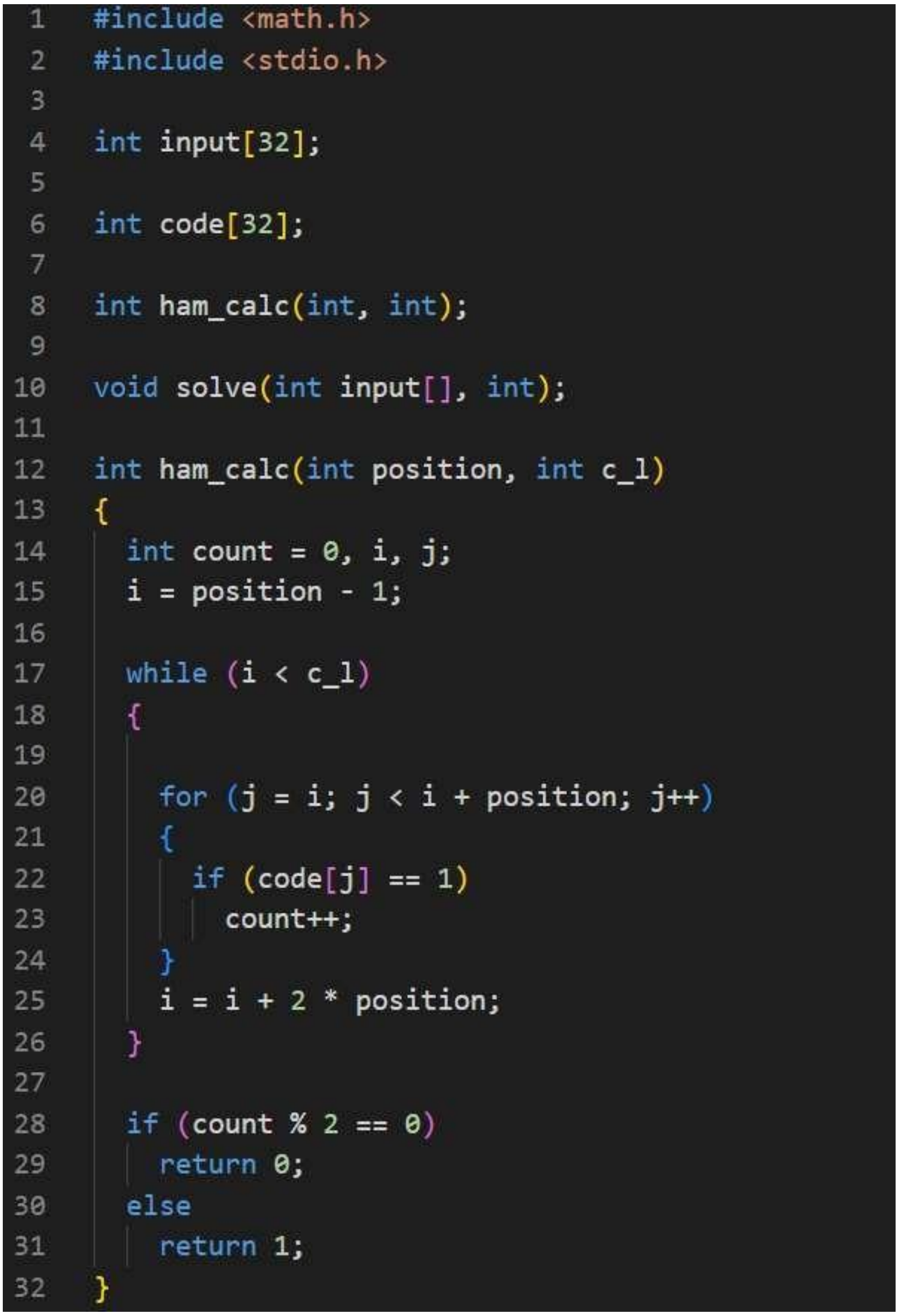
FOR EDUCATIONAL USE

· It reduces loss of data and keeps buffers in control	· It is used to detect and correct the occurred error
· Eg: Stop & Wait protocol	· Eg: Stop & Wait ARQ, Sliding window ARQ

2) Explain in breif the two types of error control mechanisms

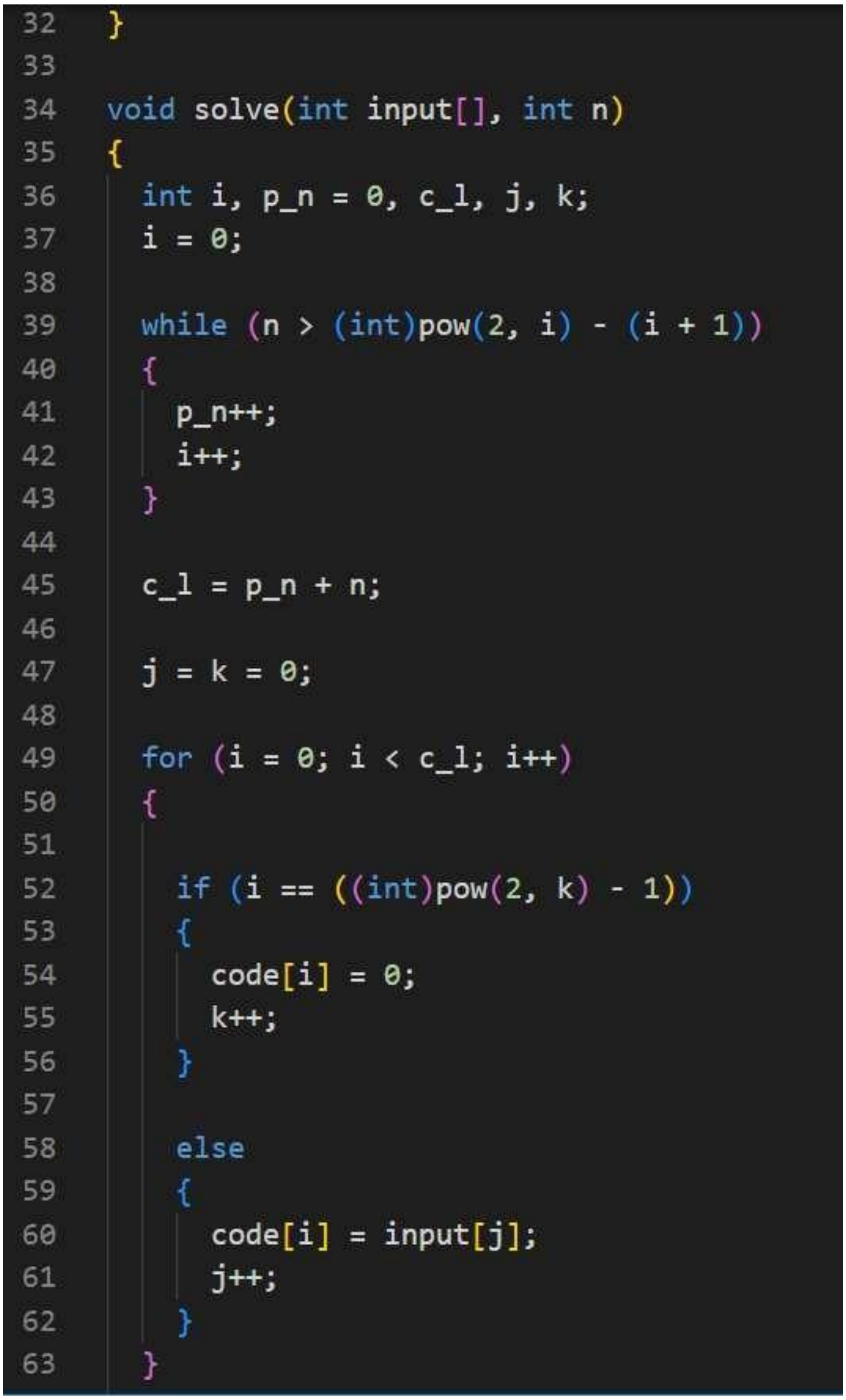
→ i) Forward Control: In this mechanism, additional redund information is also transmitted along with the data. This helps the receiver to detect & determine the location of the error in the transmitted data

ii) Feedback Control: In this along with each stream, a little additional information is provided for the detection of error. The receiver, in the technique perform no error correction. If ~~the~~ an error is detected the data is retransmitted

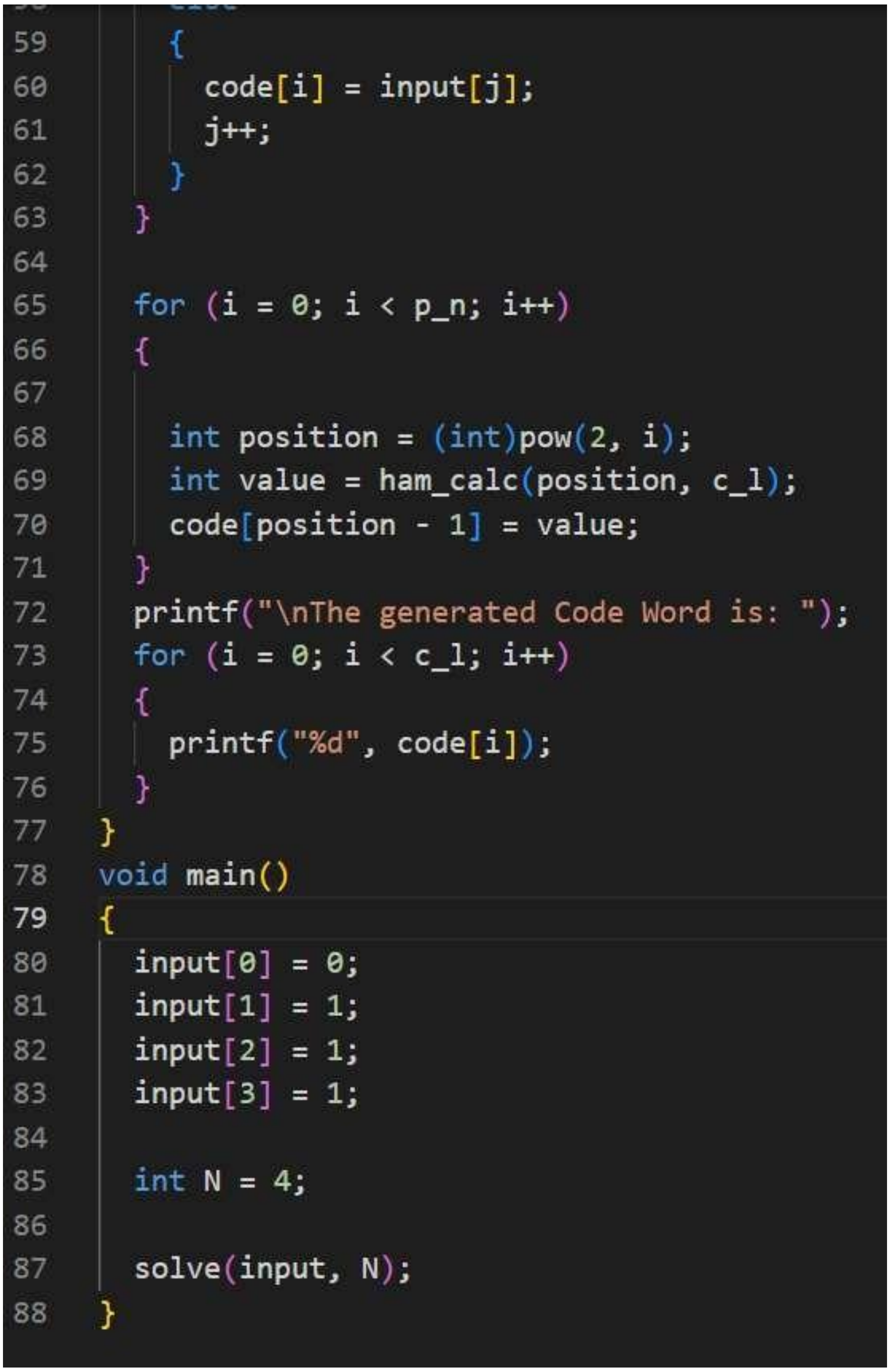


```c
#include <math.h>
#include <stdio.h>

int input[32];

int code[32];

int ham_calc(int, int);

void solve(int input[], int);

int ham_calc(int position, int c_l)
{
  int count = 0, i, j;
  i = position - 1;

  while (i < c_l)
  {

    for (j = i; j < i + position; j++)
    {
      if (code[j] == 1)
        count++;
    }
    i = i + 2 * position;
  }

  if (count % 2 == 0)
    return 0;
  else
    return 1;
}
```

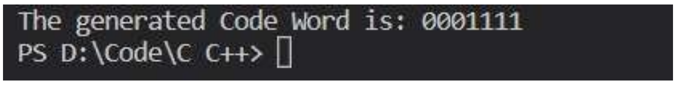
v



```c
}

void solve(int input[], int n)
{
  int i, p_n = 0, c_l, j, k;
  i = 0;

  while (n > (int)pow(2, i) - (i + 1))
  {
    p_n++;
    i++;
  }

  c_l = p_n + n;

  j = k = 0;

  for (i = 0; i < c_l; i++)
  {

    if (i == ((int)pow(2, k) - 1))
    {
      code[i] = 0;
      k++;
    }

    else
    {
      code[i] = input[j];
      j++;
    }
  }
```



```c
        {
            code[i] = input[j];
            j++;
        }
    }

    for (i = 0; i < p_n; i++)
    {

        int position = (int)pow(2, i);
        int value = ham_calc(position, c_l);
        code[position - 1] = value;
    }
    printf("\nThe generated Code Word is: ");
    for (i = 0; i < c_l; i++)
    {
        printf("%d", code[i]);
    }
}
void main()
{
    input[0] = 0;
    input[1] = 1;
    input[2] = 1;
    input[3] = 1;

    int N = 4;

    solve(input, N);
}
```



```
The generated Code Word is: 0001111
PS D:\Code\C C++> 
```