

Assignment 2

Introduction to programming in C

Question 1

Parity Checker

You are given a sequence of bits (1's and 0's). The sequence is said to have even parity if and only if the number of 1's in the sequence is even.

Write a C program to that outputs 1 if the sequence has even parity and 0 otherwise.

Input

A sequence of bits (0's and 1's) ending with a -1. (Note : -1 is not a part of input. It only signifies that input has ended)

Output

1 if the number of ones in the sequence is even. 0 if the number of ones in the sequence is odd.

Solution

```
1 #include <stdio.h>
2
3 int main() {
4     int bit;
5     int count = 0;
6
7     // Read bits until -1 is encountered
8     while (1) {
9         scanf("%d", &bit);
10
11         // Check for end of input
12         if (bit == -1) {
13             break;
14         }
15
16         // Count the number of 1's
17         if (bit == 1) {
18             count++;
19         }
20     }
```

```
21
22 // Check if the count of 1's is even
23 if (count % 2 == 0) {
24     printf("1"); // Even parity
25 } else {
26     printf("0"); // Odd parity
27 }
28
29 return 0;
30 }
```

Question 3

Count the Number of 0's Between the First and Last 1

You are given a binary sequence.

Write a C program to count the number of 0's between the first and last 1 in the sequence.

Input

A sequence of bits (0's and 1's) ending with a -1. (Note : -1 is not a part of input. It only signifies that input has ended)

Output

The number of 0's Between the First and Last 1 in the sequence.

Note : Make no assumptions about the data in the sequence. For instance if there is no starting and ending 1 (say the sequence is all 0), you have to output 0.

Solution

```
1 #include <stdio.h>
2
3 int main() {
4     int num, count = 0, count2 = 0;
5
6     // Read until first 1
7     do{
8         scanf("%d", &num);
9     } while(num==0);
10
11    // Only zeroes in the list
12    if(num == -1)
13        printf("0");
14
15    // Read rest of input
16    do{
17        scanf("%d", &num);
18        if(num == 0)
19            // Increment number of 0's seen after last 1
20            count2++;
21        else if(num == 1){
22            /*1 detected. count2 is added to count,
23            and is reset to 0*/
24            count += count2;
25            count2 = 0;
26        }
27    } while(num != -1);
28
29    printf("%d", count);
30    return 0;
31 }
```