

PPS9

Q1

Aim:

Given an array of size N, write a C program using pointer arithmetic to find the sum of the digits of all array elements which contains even number of 1's in their equivalent binary representation. If there is no such element with even number of 1's, print -1.

Procedure:

Input:

A number, 'n'

'n' array elements

Output:

Sum of digits of all elements from array that satisfies the condition else -1

Algorithm:

Step 1: Declare 'binary', 'eveodd' and 'digitsum' function with return type 'int' and argument 'int n'

Main Function

Step 1: Initialise integer variables 'n' and 'i' and read 'n'.

Step 2: Initialise and read an integer array of size n, 'arr'.

Step 3: Initialise an integer pointer variable, 'ptr', and assign 'arr' to it.

Step 4: Initialise 'sum' to 0, declare 'r'

Step 5: For each element of the array

 Step A: Pass element in 'binary function' using pointer reference

 Step B: Pass the binary number obtained in 'eveodd function' and store the return value in 'r'

 Step C: If r is equal to 1

 Then pass the element in 'digitsum function' using pointer reference and add the return value to 'sum'

 Step D: Increment 'ptr' by 1

Step 6: If 'sum' is equal to 0

 Print -1

 Else

 Print 'sum'

Step 7: Return 0

Binary Function

Step 1: Initialise 'i' to 0, 'b' to 1 and declare 'r'

Step 2: While $n > 0$

Step A: $r = n \% 2$

Step B: $b += r * i$

Step C: $i *= 10$

Step D: $n /= 2$

Step 3: Return b

Eveodd Function

Step 1: Initialise 'c' to 0 and declare 'r'

Step 2: While $n > 0$

Step A: $r = n \% 10$

Step B: If r is equal to 1

Increment c by 1

Step C: $n /= 10$

Step 3: If c is even

Return 0

Else

Return 1

Digitsum Function

Step 1: Initialise 's' to 0

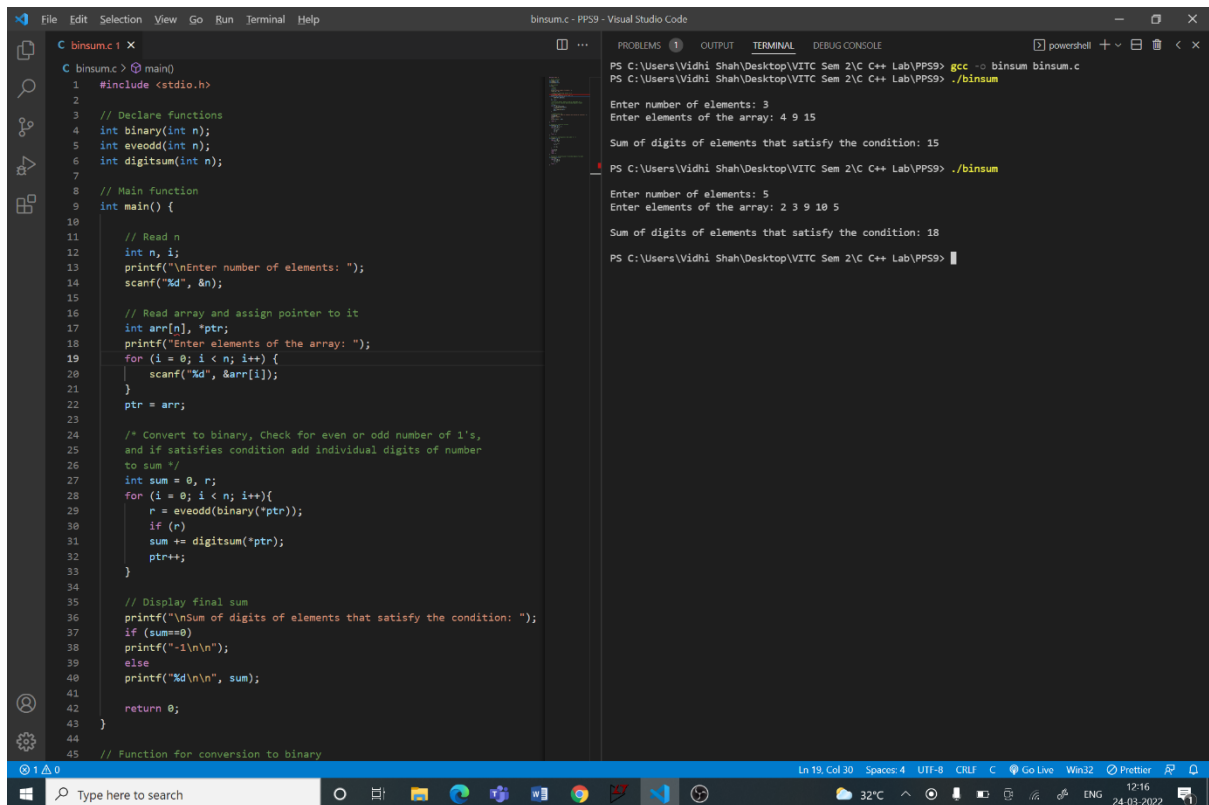
Step 2: While $n > 0$

$s += n \% 10$

$n /= 10$

Step 3: Return s

Code:



```
File Edit Selection View Go Run Terminal Help
binsum.c - PPS9 - Visual Studio Code

C binsum.c X
C binsum.c > main()
1 #include <stdio.h>
2
3 // Declare functions
4 int binary(int n);
5 int eveodd(int n);
6 int digitsum(int n);
7
8 // Main function
9 int main() {
10
11     // Read n
12     int n, i;
13     printf("\nEnter number of elements: ");
14     scanf("%d", &n);
15
16     // Read array and assign pointer to it
17     int arr[n], *ptr;
18     printf("Enter elements of the array: ");
19     for (i = 0; i < n; i++) {
20         scanf("%d", &arr[i]);
21     }
22     ptr = arr;
23
24     /* Convert to binary, Check for even or odd number of 1's,
25     and if satisfies condition add individual digits of number
26     to sum */
27     int sum = 0, r;
28     for (i = 0; i < n; i++){
29         r = eveodd(binary(*ptr));
30         if (r)
31             sum += digitsum(*ptr);
32         ptr++;
33     }
34
35     // Display final sum
36     printf("\nSum of digits of elements that satisfy the condition: ");
37     if (sum==0)
38         printf("-1\n");
39     else
40         printf("%d\n", sum);
41
42     return 0;
43 }
44
45 // Function for conversion to binary
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> gcc -o binsum binsum.c

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> ./binsum

Enter number of elements: 3

Enter elements of the array: 4 9 15

Sum of digits of elements that satisfy the condition: 15

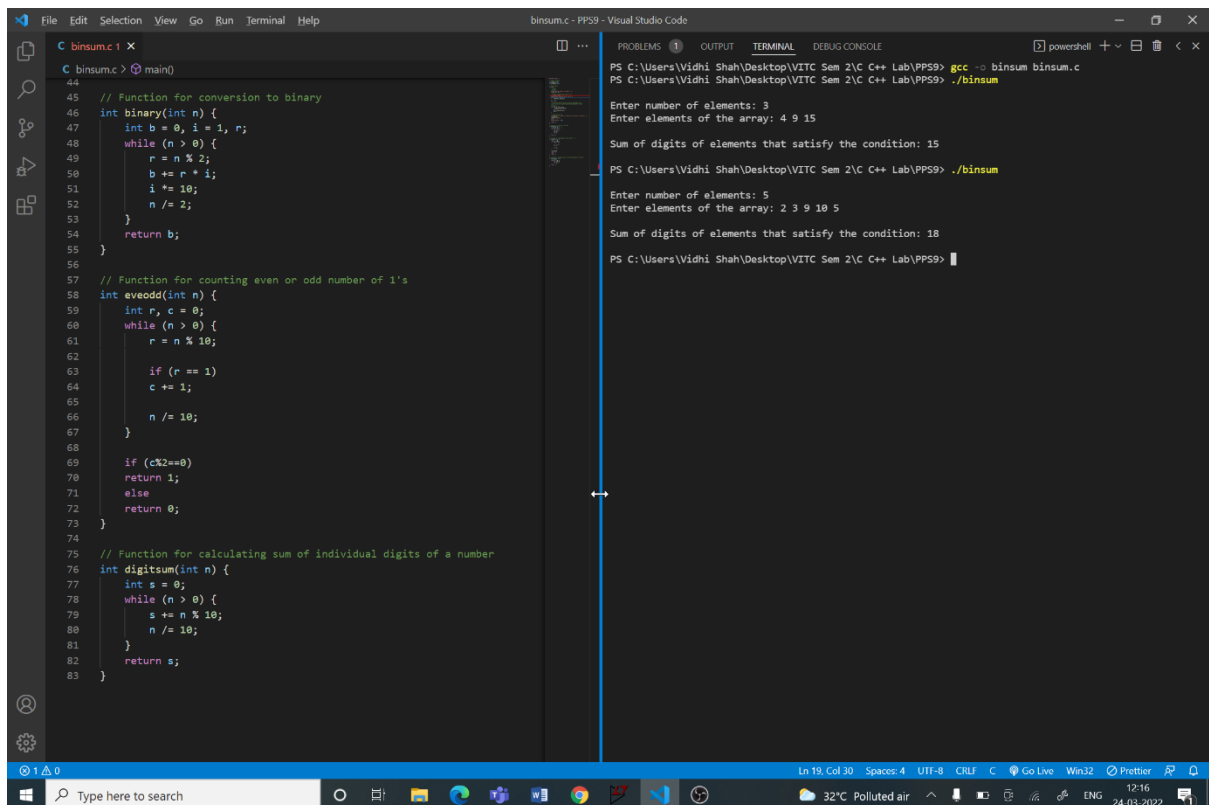
PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> ./binsum

Enter number of elements: 5

Enter elements of the array: 2 3 9 10 5

Sum of digits of elements that satisfy the condition: 18

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9>



```
File Edit Selection View Go Run Terminal Help
binsum.c - PPS9 - Visual Studio Code

C binsum.c X
C binsum.c > main()
44
45 // Function for conversion to binary
46 int binary(int n) {
47     int b = 0, i = 1, r;
48     while (n > 0) {
49         r = n % 2;
50         b += r * i;
51         i *= 10;
52         n /= 2;
53     }
54     return b;
55 }
56
57 // Function for counting even or odd number of 1's
58 int eveodd(int n) {
59     int r, c = 0;
60     while (n > 0) {
61         r = n % 10;
62         if (r == 1)
63             c += 1;
64         n /= 10;
65     }
66     if (c%2==0)
67         return 1;
68     else
69         return 0;
70 }
71
72 // Function for calculating sum of individual digits of a number
73 int digitsum(int n) {
74     int s = 0;
75     while (n > 0) {
76         s += n % 10;
77         n /= 10;
78     }
79     return s;
80 }
81
82
83 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> gcc -o binsum binsum.c

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> ./binsum

Enter number of elements: 3

Enter elements of the array: 4 9 15

Sum of digits of elements that satisfy the condition: 15

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9> ./binsum

Enter number of elements: 5

Enter elements of the array: 2 3 9 10 5

Sum of digits of elements that satisfy the condition: 18

PS C:\Users\Vidhi Shah\Desktop\VITC Sem 2\C++ Lab\PPS9>

```

#include <stdio.h>

// Declare functions
int binary(int n);
int eveodd(int n);
int digitsum(int n);

// Main function
int main() {

    // Read n
    int n, i;
    printf("\nEnter number of elements: ");
    scanf("%d", &n);

    // Read array and assign pointer to it
    int arr[n], *ptr;
    printf("Enter elements of the array: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    ptr = arr;

    /* Convert to binary, Check for even or odd number of 1's,
    and if satisfies condition add individual digits of number
    to sum */
    int sum = 0, r;
    for (i = 0; i < n; i++){
        r = eveodd(binary(*ptr));
        if (r)
            sum += digitsum(*ptr);
        ptr++;
    }

    // Display final sum
    printf("\nSum of digits of elements that satisfy the condition: ");
    if (sum==0)
        printf("-1\n\n");
    else
        printf("%d\n\n", sum);

    return 0;
}

```

```

// Function for conversion to binary
int binary(int n) {
    int b = 0, i = 1, r;
    while (n > 0) {
        r = n % 2;
        b += r * i;
        i *= 10;
        n /= 2;
    }
    return b;
}

// Function for counting even or odd number of 1's
int eveodd(int n) {
    int r, c = 0;
    while (n > 0) {
        r = n % 10;

        if (r == 1)
            c += 1;

        n /= 10;
    }

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

// Function for calculating sum of individual digits of a number
int digitsum(int n) {
    int s = 0;
    while (n > 0) {
        s += n % 10;
        n /= 10;
    }
    return s;
}

```