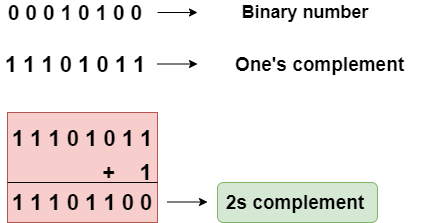
What is the 2s complement in C?

The 2s complement in C is generated from the 1s complement in C. As we know that the 1s complement of a binary number is created by transforming bit 1 to 0 and 0 to 1; the 2s complement of a binary number is generated by adding one to the 1s complement of a binary number.

In short, we can say that the 2s complement in C is defined as the sum of the one's complement in C and one.



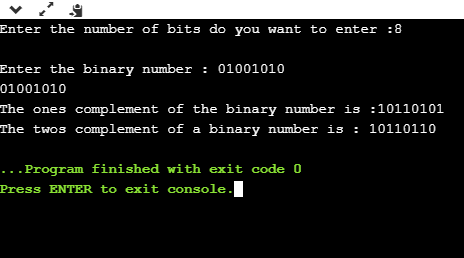
In the above figure, the binary number is equal to 00010100, and its one's complement is calculated by transforming the bit 1 to 0 and 0 to 1 vice versa. Therefore, one's complement becomes 11101011. After calculating one's complement, we calculate the two's complement by adding 1 to the one's complement, and its result is 11101100.

Let's create a program of 2s complement.

1. #include <stdio.h>
2. **int** main()
3. {
4. **int** n;  // variable declaration
5. printf("Enter the number of bits do you want to enter :");
6. scanf("%d",&n);
7. **char** binary[n+1];  // binary array declaration;
8. **char** onescomplement[n+1]; // onescomplement array declaration
9. **char** twoscomplement[n+1]; // twoscomplement array declaration
10. **int** carry=1; // variable initialization
11. printf("\nEnter the binary number : ");
12. scanf("%s", binary);
13. printf("%s", binary);
14. printf("\nThe ones complement of the binary number is :");
16. // Finding onescomplement in C
17. **for**(**int** i=0;i<n;i++)
18. {
19. **if**(binary[i]=='0')
20. onescomplement[i]='1';
21. **else** **if**(binary[i]=='1')
22. onescomplement[i]='0';
23. }
24. onescomplement[n]='\0';
25. printf("%s",onescomplement);

28. printf("\nThe twos complement of a binary number is : ");
30. // Finding twoscomplement in C
31. **for**(**int** i=n-1; i>=0; i--)
32. {
33. **if**(onescomplement[i] == '1' && carry == 1)
34. {
35. twoscomplement[i] = '0';
36. }
37. **else** **if**(onescomplement[i] == '0' && carry == 1)
38. {
39. twoscomplement[i] = '1';
40. carry = 0;
41. }
42. **else**
43. {
44. twoscomplement[i] = onescomplement[i];
45. }
46. }
47. twoscomplement[n]='\0';
48. printf("%s",twoscomplement);
49. **return** 0;
50. }

**Output**



Analysis of the above program,

* First, we input the number of bits, and it gets stored in the '**n**' variable.
* After entering the number of bits, we declare character array, i.e., **char binary[n+1],** which holds the binary number. The '**n**' is the number of bits which we entered in the previous step; it basically defines the size of the array.
* We declare two more arrays, i.e., **onescomplement[n+1]**, and **twoscomplement[n+1].** The **onescomplement[n+1]** array holds the ones complement of a binary number while the **twoscomplement[n+1]** array holds the two's complement of a binary number.
* Initialize the **carry** variable and assign 1 value to this variable.
* After declarations, we input the binary number.
* Now, we simply calculate the one's complement of a binary number. To do this, we create a **loop** that iterates throughout the binary array, **for(int i=0;i<n;i++)**. In for loop, the condition is checked whether the bit is 1 or 0. If the bit is 1 then **onescomplement[i]=0**else **onescomplement[i]=1**. In this way, one's complement of a binary number is generated.
* After calculating one's complement, we generate the 2s complement of a binary number. To do this, we create a **loop** that iterates from the last element to the starting element. In for loop, we have three conditions:
  + If the bit of onescomplement[i] is 1 and the value of carry is 1 then we put 0 in twocomplement[i].
  + If the bit of onescomplement[i] is 0 and the value of carry is 1 then we put 1 in twoscomplement[i] and 0 in carry.
  + If the above two conditions are false, then onescomplement[i] is equal to twoscomplement[i].