Problem 1:
C program to convert a binary number to octal number.
Input:
101
Output:
5
Input:
11010
Output:
32
Test Case:
---
1. Valid Input:
a) Only number consisting of 0s and 1s will be given as input

2. Invalid inputs:
a) Decimal
b) Fraction
c) String
d) Two or more command line arguments
e) Negative number

3. You should generate output as follows:
a) For right output print just the actual Octal Value to STDOUT without any other text.
b) If any error: print 'ERROR' to the STDOUT without any other text.

Program 2:
Given an array of integers, sort the first half of the array in ascending order and second half in descending order. Take input from STDIN by scanf().
Examples:
Input : arr[] = {5, 2, 4, 7, 9, 3, 1, 6, 8}
Output : arr[] = {1, 2, 3, 4, 9, 8, 7, 6, 5}
Input : arr[] = {1, 2, 3, 4, 5, 6}
Output : arr[] = {1, 2, 3, 6, 5, 4}

Program 3:
A prime number is a whole number greater than 1, whose only two whole-number factors are 1 and itself. The first few prime numbers are 2, 3, 5, 7, 11 and 13.
Given two integers, print the sum of all prime numbers between n1 and n2 (both inclusive).
Input
The first line of the input contains an integer t, number of test cases next t lines follow with each line containing two integers n1 and n2
Output
For each test case print the answer in a new line.
Example
Input:

2
1 6
6 6
Output:
10
0
Test Cases:
---
1. Valid Input:
a) Only integer will be given as input.
Constraints:
1 < t <= 10
0 < n1 <= n2 <= 100

2. Invalid inputs:
a) Negative number (t, n1, or n2)
b) Fraction
c) String

3. You should generate output as follows:
a) For right output print just the sum to STDOUT without any other text.
b) If any error: print 'ERROR' to the STDOUT without any other text.

Program 4:
Remove duplicates character from a given string.
Input String:
engineering
Output String:
engir
Test Cases:
---
1. Valid Input:
a) Only string of character will be given as input.

2. Invalid inputs:
a) No command line arguments
b) Two or more command line arguments
c) Integer
d) Fraction

3. You should generate output as follows:
a) For right output print just the string to STDOUT without any other text.
b) If any error: print 'ERROR' to the STDOUT without any other text.

Program 5
C Program to find sum of series 1 + 1/2 + 1/3 + 1/4 +... + 1/n. The value n is positive integer

passed to the program as the first command line parameter. Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text. Scientific format (such as 1.00E+5) should NOT be used while printing the output. You may assume that the inputs will be such that the output will not exceed the largest possible real number that can be stored in a float type variable.

Example
Input
1
Output
1
Input
2
Output
1.5

Test Cases:
---
1. VALID INPUTS:
a) Only positive integer will be given as input through command line argument.

2. INVALID INPUTS:
a) No command line argument.
b) More than 1 command line arguments
c) String
d) Fraction
e) Negative number as input argument

3. OUTPUT:
a) Write the output to stdout formatted as a floating point number rounded to EXACTLY 2 decimal precision WITHOUT any other additional text.
For example
[./a.out 2] => 1.5
b) In case of invalid input print 'ERROR' to the STDOUT without any other additional text and terminate.