American Computer Science League

2022-2023 ● Contest 2: Binary Counting ● Senior Division

PROBLEM: Given a string of characters found on the keyboard, convert each character in the string to the binary equivalent of its ASCII code. In the resulting concatenated string, search for the increasing sequence of binary numbers starting with 0, 1, 10, 11, … until a number cannot be found anywhere in the string. Look from the start of the string. If the binary number is found, remove that occurrence of the binary number from the string. Then look from the end of the string. If the binary number is found, remove that occurrence of the binary number from the string. Once a number in the sequence cannot be found, convert the string to an octal number without leading 0s. Then repeat this same process with a sequence of base 8 numbers 0, …, 7, 10, 11, …. Output the last octal number, converted to base 10, that can successfully be found. If 0 cannot be found, output -1.

EXAMPLE: For the string "Roses are red.", convert it to a concatenated string of binary numbers using each character's ASCII code as follows:

Char	ASCII	Binary
R	82	01010010
o	111	01101111
s	115	01110011
e	101	01100101
s	115	01110011
sp	32	00100000
a	97	01100001

Char	ASCII	Binary
r	114	01110010
e	101	01100101
sp	32	00100000
r	114	01110010
e	101	01100101
d	100	01100100
.	46	00101110

Now search for binary numbers beginning with 0 in the following string:

0<u>1</u>010010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 0010111<u>0</u>

Remove the 0 from both ends so the string becomes:

<u>1</u>010010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 0010111

Remove the 1 from both ends so the string becomes:

0<u>10</u>010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 00<u>10</u>11

Remove 10 from both ends so the string becomes:

0010 0<u>11</u>01111 01110011 01100101 01110011 00100000 01100001

01110010 01100101 00100000 01110010 01100101 01100100 0011

Remove 11 from both ends so the string becomes:

0010 001111 01110011 01100101 01110011 00100000 01100001

01110010 01100101 00100000 01110010 01100101 01100100 00

After removing 100 from both ends as shown above, the process continues until the final string becomes:

000011000001100001001001000000001000100

The string 1101 cannot be found in this final string. The binary numbers 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, and 1100 were found and deleted from one or both sides of the string, but not 1101.

The resulting string converted to an octal number is 603022200104. Repeating the same process, the sequence of strings is 603022200104, 6302220014, 63022004, 6302004, 602004, 60200 so the last octal number that can be found is 4 since 5 cannot be found in the string.

INPUT: A string containing any character that can be found on the keyboard. The string will be fewer than 200 characters.

OUTPUT: Once a number in the sequence cannot be found, output the last octal number, converted to base 10, that can be found in the string after handling all deletions as explained above. If 0 cannot be found, output -1.

SAMPLE INPUT:

```
1. Roses are red.
2. A is for Alpha; B is for Bravo; C is for Charlie.
3. A stitch in time saves nine.
4. 1, 2: Buckle my shoe! 3, 4: Shut the door!
5. The quick brown fox jumped over the lazy dogs.
```

SAMPLE OUTPUT:

```
1. 4
2. 9
3. 8
4. 6
5. 5
```