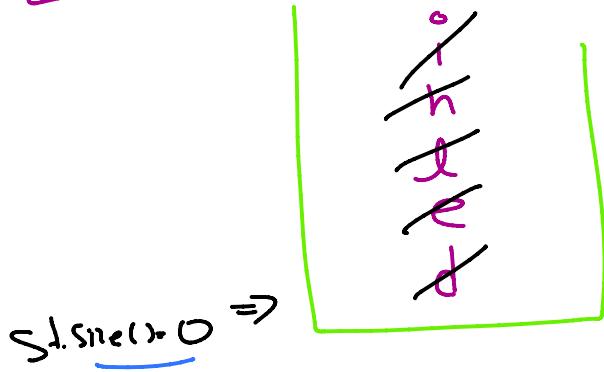


→ - Demo

$\text{ans} += \text{st.top()};$

LIFO \Rightarrow Reverse



ans = st.top();

city = "Delhi"

ans = ihled

google youtube twitter



LIFO

given \Rightarrow i.like.this.program.very.much

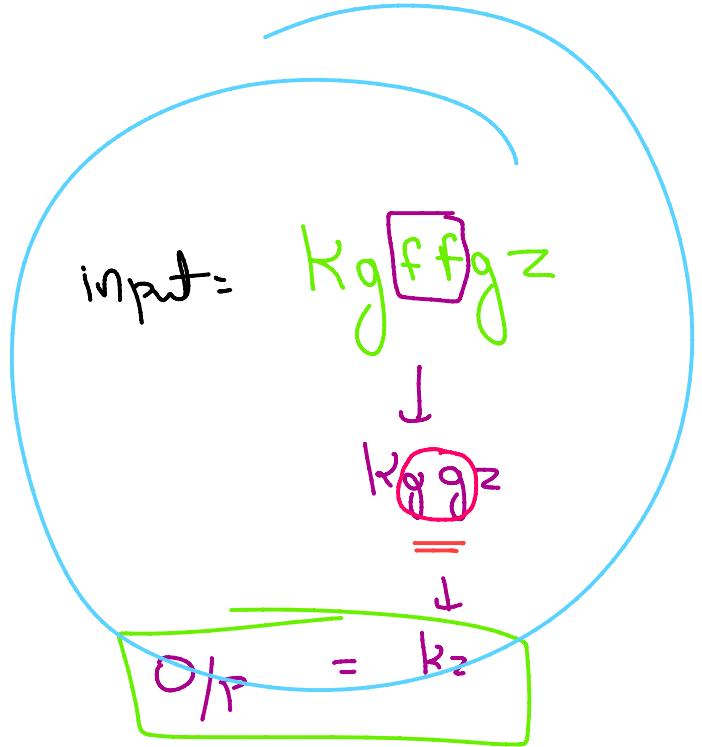
O/p \Rightarrow much.very.program.this.like.i

LIFO

input = geek

O/P = kz

g e e k
^ ^
i-1 i



kgf + gkz
^ ^
i-1 i

kgf + gkz
= kz

last char

The diagram shows a stack represented by a blue circle containing the string "kgf + gkz". A green box highlights the prefix "gf". A red box highlights the suffix "gkz". An arrow points from the stack to a box labeled "O/P = kz".

last character
g e e k
== ↑

$\text{ans} = \underline{\underline{g^k}}$

derivation

geek

$\text{ans} = \underline{\underline{g^e k}}$

y K geek z

$\text{ans} = \boxed{z^y}$

The diagram illustrates a sequence of symbols, likely representing a string of characters or a specific pattern. The sequence consists of the symbol 'Yggdrasil' repeated multiple times, with each symbol rendered in black ink. To the left of this sequence, there is handwritten text in pink ink that appears to be a label or identifier. Below the main sequence, a green L-shaped bracket is drawn, spanning the width of several symbols. To the right of the bracket, there is handwritten text in pink ink, which includes the word 'ans' (answers) and 'ser.' (series).

o b b a c o ↑

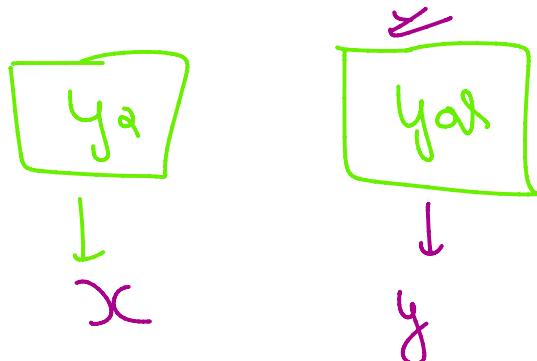
a
c

ac

dvw = C2

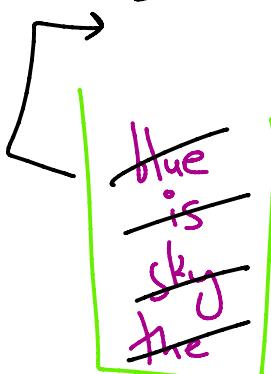
Strings are immutable in java.

Ans = "de" Ans = "dez"



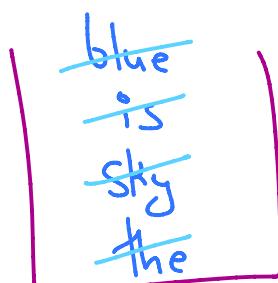
Q. Reverse words in the given String.

str = str



str = "the sky is blue"

O/p = "blue is sky the"



str = "the sky is blue"



Stack

.

Q. Reverse words in a given String.

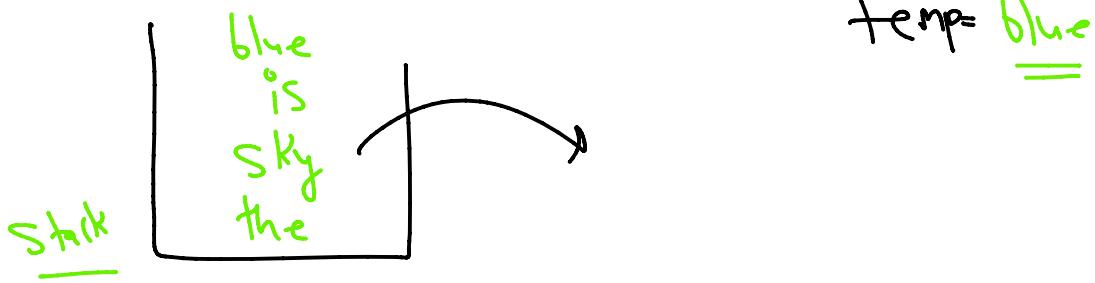
String → _
 ↑

String = " __ the _ sky - - is - blue - "

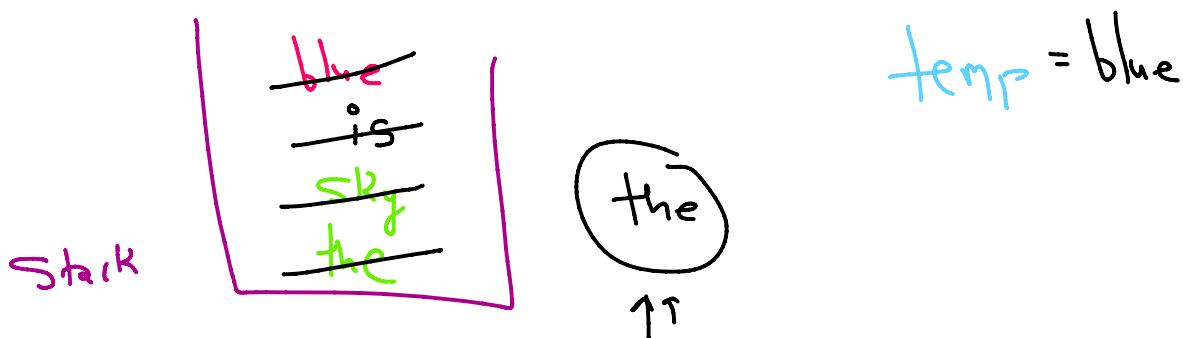
⇒ "the _ sky - is _ blue"

| blue
 |
 |

temp = blue



String = " __ the _ sky - - is - - blue" ↑



ans = blue_is_sky_the

while (st.size() > 0)

{

word = st.top();
st.pop();

if (st.size() > 0)

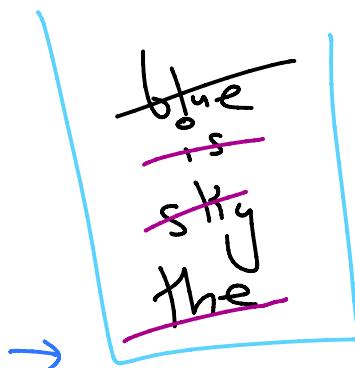
{

ans += word + " ";

```

        }
    } else
    {
        ans += work;
    }
}

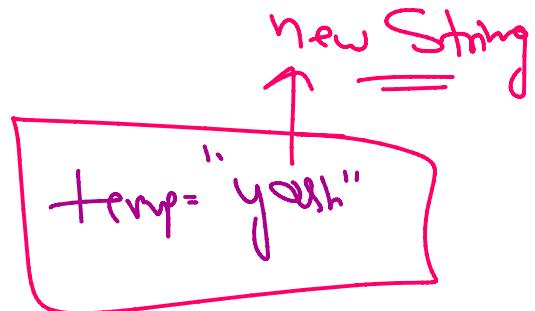
```



$\text{ans} = \text{blue_is_sky_the}$

$\text{temp} = "yal"$

$\text{temp} = "h"$



Remove adjacent duplicate \Rightarrow 2 Same adjacent characters then remove them.

$k_1 f f g z = k_2$

$$Kg \boxed{f f f} g z = Kz$$

$$Kg \boxed{\cancel{f f f}} g z$$

$$\underline{Kggz} = (Am)$$

$$K = 3$$

K adjacent

if 3 adjacent
characters are
same then
2Q more
them.

$$ygg \boxed{f f f} g z$$

$$\boxed{K = 3}$$

$$y \boxed{\cancel{g g g}} z$$

y z (Ans.)

D

Q. Remove adjacent duplicate (K consecutive)
(part-II)

\Rightarrow "deeeedbbcccbdaaa"

$$K=3$$

deeeedbbcccbdaaa

$$K=3$$

ddbbcccbdaaa

ddbbbdaaa

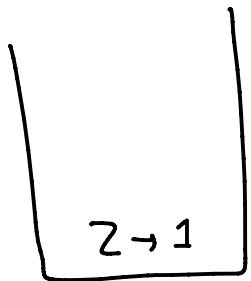
dddaaa

daaa

\Rightarrow 2dddccccd

$$K=3$$

\Rightarrow ~~z d d c c c d~~

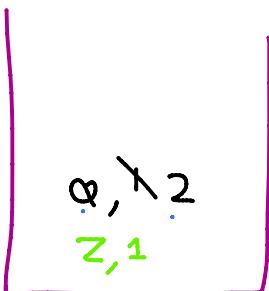


Stack <pair>

pair \rightarrow (char, int)

pair → (`char`, `int`)

~~g e e e g g f f f c c c f a a a~~ → 2a°
(Ans)



$$\text{Ans} = \frac{0.2}{\downarrow} Z$$

Stack

Zoo

Pair <char, int>

→ ←
if

Se cal

ch

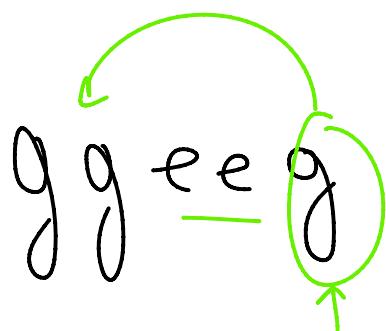


Pair

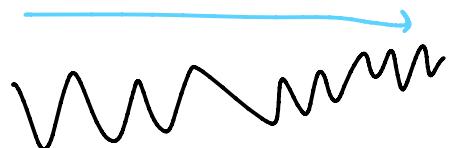
char, int

Key

value



$k = 3$



$T(): \rightarrow O(n)$ ✓

$S(): \rightarrow \underline{O(n)}$

