Strings

▼ Intro

Strings can be defined as group of characters

- In JAVA Strings are immutable
- String str = “abcd”
- .length() is used to find the length of a string . It is used as a function in case of strings
- Concatenation : can be done using +
- .charAt(0) : is used to find the character at any index value. To print the complete string we can use .carAt(i) in a loop

▼ Palindrome

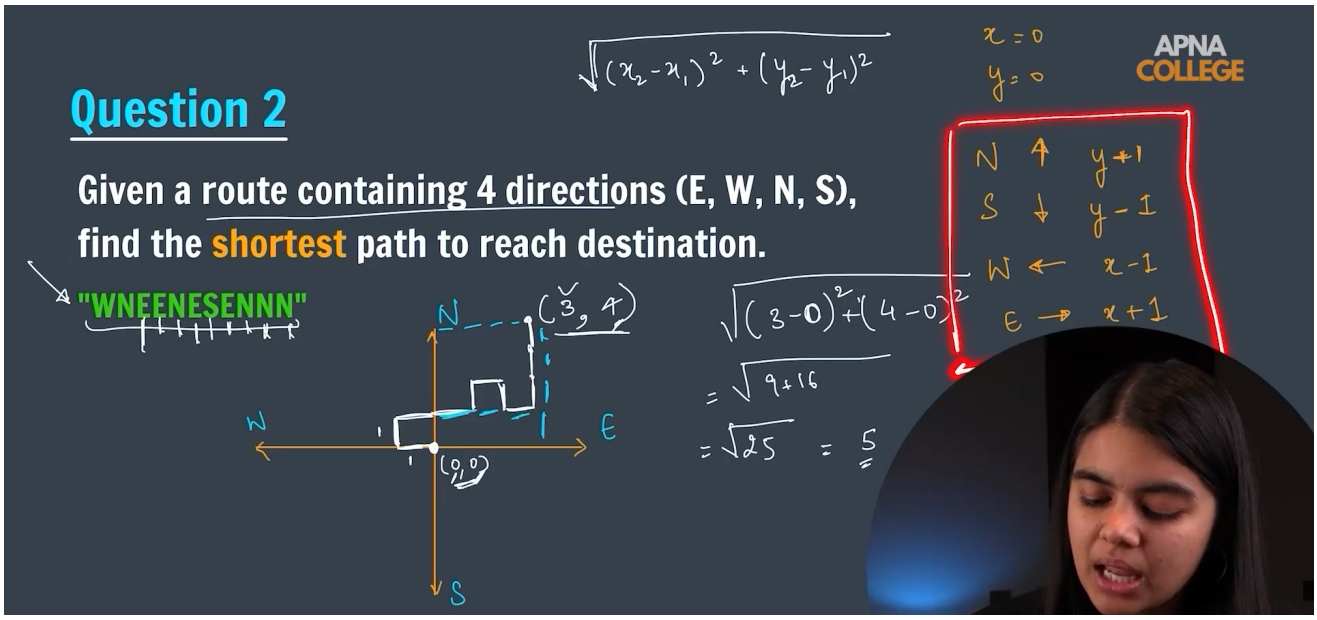
Check if the string is a palindrome

- A string which is same to read no matter if it is started reading from the front or from the back e.g. noon , madam
  - Logic : Compare First character to the last , then the second char to the second last char and similarly all char upto the middle of the string
  - Loop would be performed upto half of the length of the loop
  - comparing str(i) = str (n - i - 1)
  - n is the length of the string starting form 0
  - Time Complexity : Linear O (n)

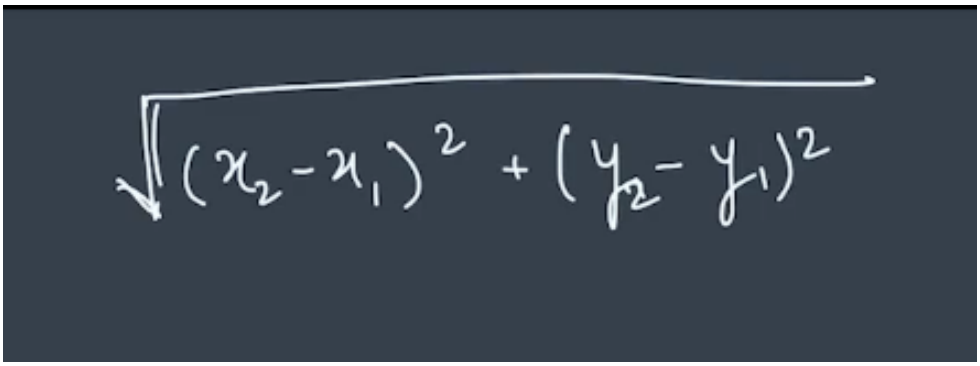
▼ Shortest Path

Given a route containing 4 directions (E, W, N, S),
find the shortest path to reach destination.

"WNEENESENNN”



- We move one unit based on the Letter of the String denoting the direction
  - N = y + 1
  - S = y - 1
  - W = x + 1
  - E = x + 1
- Formula to find the Shortest Path :



- Time Complexity : Linear O (n)

▼ Comparing Strings

- Concept of Interning in JAVA :
  - if a string “hello’ already stored in for e.g. by a name Str1 , then
  - if we create another string “hello” as Str2 ,

- the new string would not take up new space rather it would just point towards Str1

- To create a new string use new key word as Str3 = new string (“Hello”)

```
if(s1.equals(s3)) {
System. out. println("Strings are equal");
else
Sys em. out. println("Strings are not equal")
```

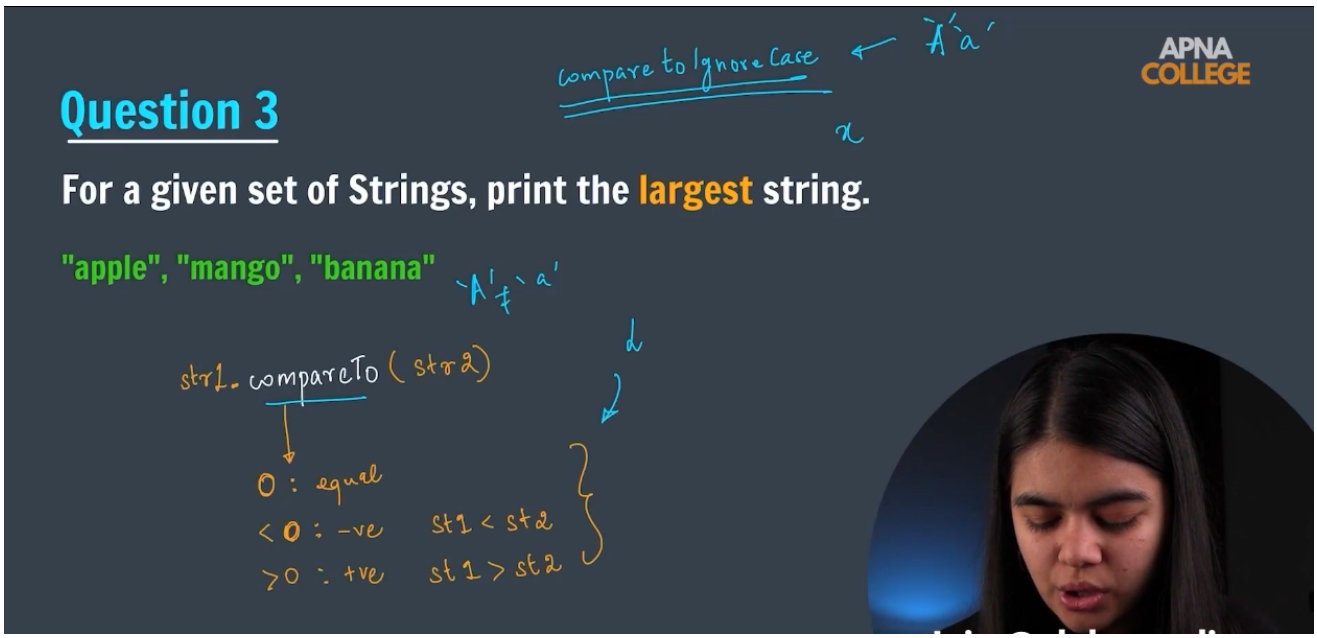
.equal function only checks the values

▼ Sub Strings

```
public static void main(String args[])
// Substring
System.out.println(str.substring (0,5))
```

▼ Largest String

Lexographically arranging order




```
public static void main(String args[]) {
String fruits [] = {"apple", "mango","banana"};\
```

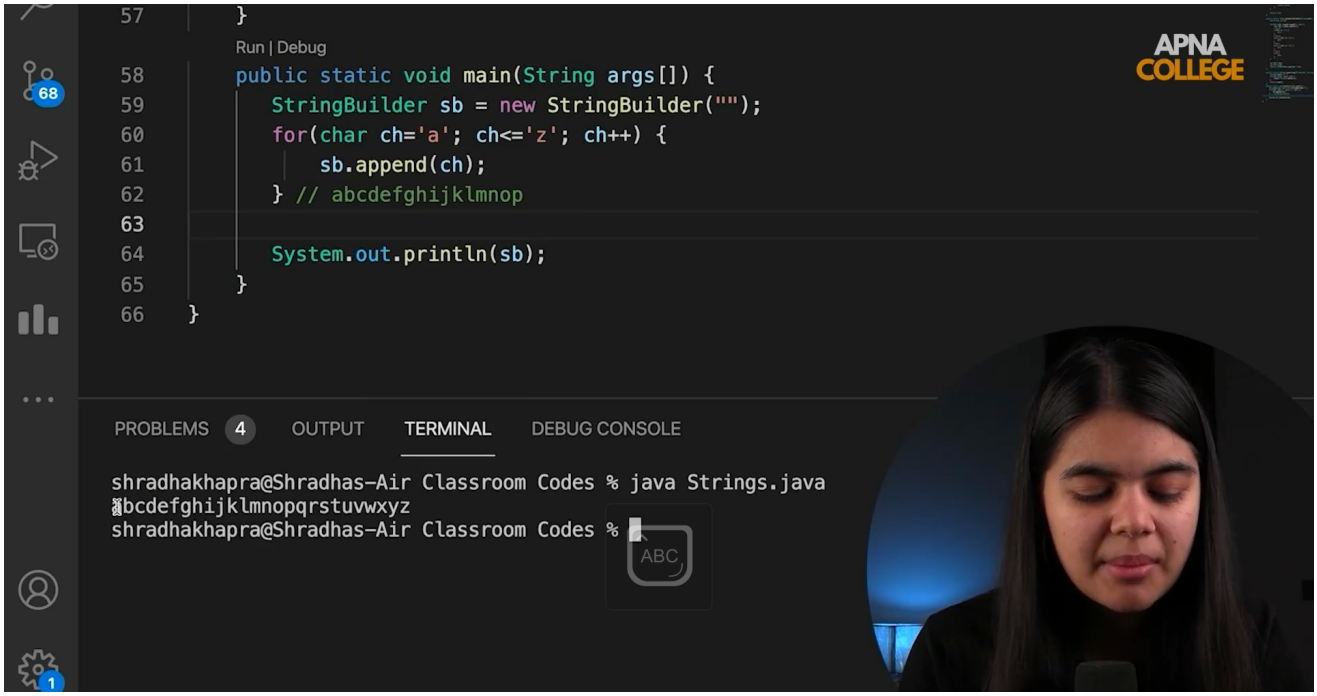
```
String largest = fruits [0] ;
for(int i=l; i<fruits.length; i++) {
  if (largest.compareTo(fruits[i] < 0) {
    largest = fruits [i];
  }
}
System.out.println(largest);
```

▼ String Builders

String Builders are used because the traditional way of adding some chars to an already existing String is a very inefficient way which could even add up to a time complexity of O (n 2sq)

- Therefore we use string builders to add char by using .append(Ch)
- O (n)

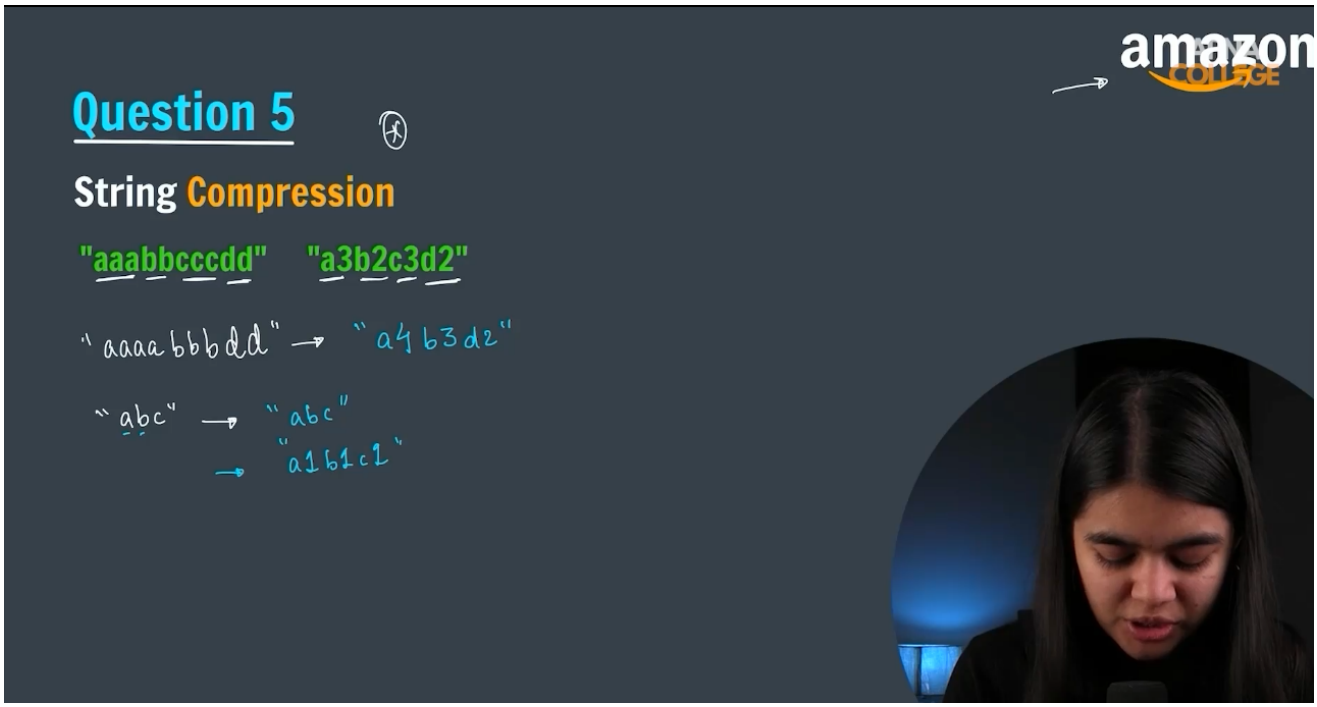
```
public static void main(String args[]) {

StringBuilder sb = new StringBuilder("");
for(char ch='a'; ch <='z' ; ch++ ){
  sb.append(ch);
// abcdefghijklmnop

System.out.println(sb) ;
```



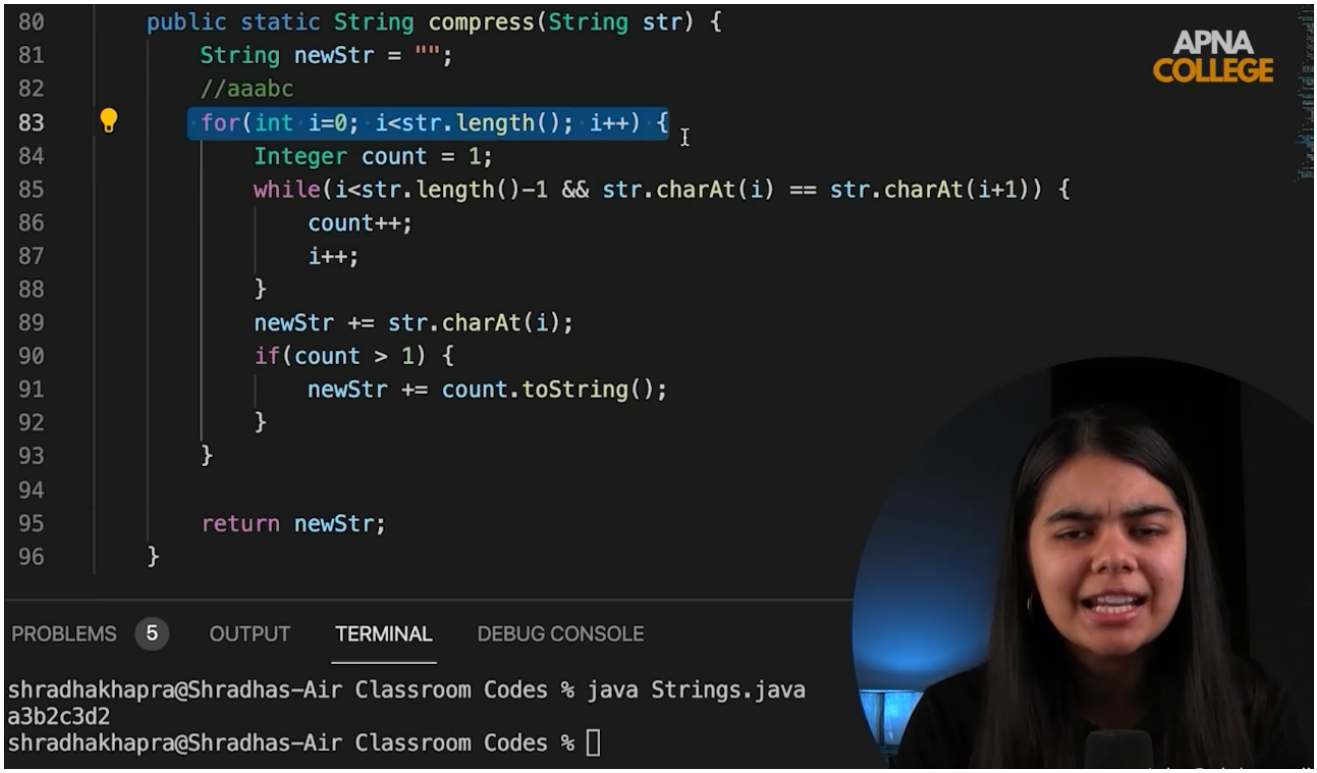
$$On^2$$

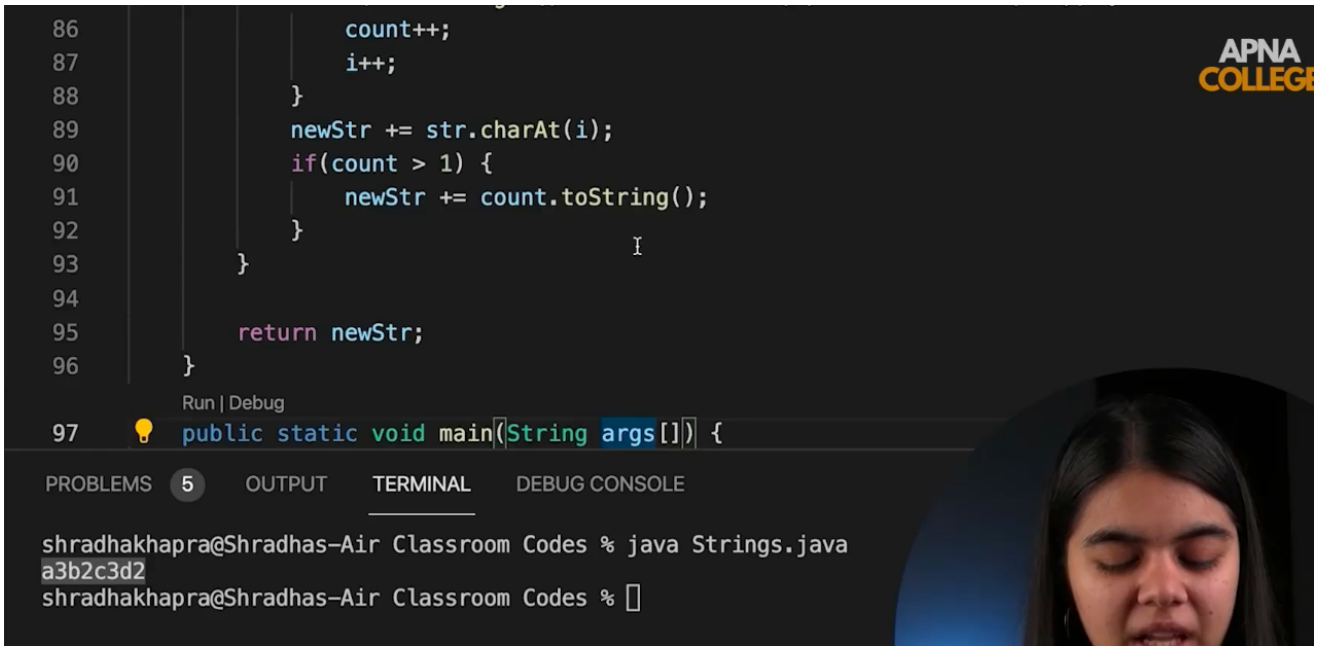
▼ Convert each first letter to Uppercase

▼ String Compression



- Initializing a count variable




APNA COLLEGE
86 count++;
87 i++;
88 }
89 newStr += str.charAt(i);
90 if(count > 1) {
91 newStr += count.toString();
92 }
93 }
94
95 return newStr;
96 }
Run | Debug
97 public static void main(String args[]) {
PROBLEMS 5
OUTPUT
TERMINAL
DEBUG CONSOLE
shradhakhapra@Shradhas-Air Classroom Codes % java Strings.java
a3b2c3d2
shradhakhapra@Shradhas-Air Classroom Codes %