We are given 3 strings: str1, str2, and str3. Str3 is said to be a shuffle of str1 and str2 if it can be formed by interleaving the characters of str1 and str2 in a way that maintains the left to right ordering of the characters from each string. For example, given str1="abc" and str2="def", str3="dabecf" is a valid shuffle since it preserves the character ordering of the two strings. So, given these 3 strings write a function that detects whether str3 is a valid shuffle of str1 and str2.

```
import java.util.*;
public classfindShortestSubString{
	public static void main(String[] args)
	{
	Scanner s = new Scanner(System.in);
	System.out.println("Enter the String1 : ");
	String str1=s.next();
	System.out.println("Enter the String2 : ");
	String str2=s.next();
	System.out.println("Enter the String1 : ");
	String str3=s.next();
	intj=0,k=0;
	for(inti=0;i<str3.length();i++)
	{
	if(j<str1.length() &&str3.charAt(i)==str1.charAt(j))
	{
	j++;
	}
	elseif(k<str2.length() &&str3.charAt(i)==str2.charAt(k))
	{
	k++;
	}
	else
	{
	break;
	}
	}
	if(j==str1.length() &&k==str2.length())
	{
	System.out.println("Valid Shuffle");
	}
	else
	{
	System.out.println("Invalid Shuffle");
	}
	}
	}
```

# OUTPUT: