**LC#344:REVERSE STRING**

class Solution {

public void reverseString(char[] s) {

int l = 0 , r = s.length-1;

while(l<r)

{

char temp = s[l];

s[l++] = s[r];

s[r--] = temp;

}

}

}

**LC #345: REVERSE VOWELS IN A STRING**

class Solution {

boolean isVowel(char c)

{

c= Character.toLowerCase(c);

if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')

return true;

return false;

}

public String reverseVowels(String s) {

int l = 0 , r = s.length()-1;

char ch[] = s.toCharArray();

while(l<r)

{

if(!isVowel(ch[l]))

l++;

else if(!isVowel(ch[r]))

r--;

else

{

char temp = ch[l];

ch[l] = ch[r];

ch[r] = temp;

l++;

r--; }}

return new String(ch);}}

**LC#151 REVERSE WORDS IN A STRING:**

class Solution {

    public String reverseWords(String s) {

        StringBuilder sb = new StringBuilder("");

        String[] words = s.trim().split("\\s+");

        for (int len = words.length-1;len>=0;len--)

        {

            sb.append(words[len]);

            if(len!=0)

                sb.append(" ");

        }

        return sb.toString().trim();

    }

}

**LC#541:REVERSE STRING II**

class Solution {

    void reverse(char[] ch ,int l ,int r){

        while(l<r)

        {

            char temp = ch[l];

            ch[l++] = ch[r];

            ch[r--] = temp;

        }

    }

    public String reverseStr(String s, int k) {

        char ch[] = s.toCharArray();

        int len = ch.length;

        for(int i = 0;i<len;i+=2\*k)

        {

            int start = i , end = Math.min(i+k-1,len-1);

            reverse(ch,start,end);// or just use two pointer while loop here

        }

        return new String(ch);

    }

}