```
# This function mainly returns LCS(str, str)
# with a condition that same characters at
# same index are not considered.
def findLongestRepeatingSubSeq( str):

	n = len(str)

	# Create and initialize DP table
	dp=[[0 for i in range(n+1)] for j in range(n+1)]

	# Fill dp table (similar to LCS loops)
	for i in range(1,n+1):
		for j in range(1,n+1):
			# If characters match and indexes are
			# not same
			if (str[i-1] == str[j-1] and i != j):
				dp[i][j] = 1 + dp[i-1][j-1]

			# If characters do not match
			else:
				dp[i][j] = max(dp[i][j-1], dp[i-1][j])


	return dp[n][n]


# Driver Program
if _name=='__main_':
	str = "aabb"
	print("The length of the largest subsequence that repeats itself is : "
		,findLongestRepeatingSubSeq(str))
```