JB23C5E13	£131 KNB233	32. St. 13. 1 to	ENB 23 CSE ST	UDENT RE	PORT	F1853C2	CSE137 KUP	823 SEL13
DET Na	TAILS Strange	(13) KN813 (2E13)	13C5E131 NUB23C	W823C5E1371	£137 £1823 (5EL)	23C5E131 LUBRS	3, YN823C5E137	5E131 LUB23654
	SHAIKH MOHAMMA	D KAIF						7
NB23 Ro	oll Number	3355	478,	CE13	.8230	21 to.	Skn	NB23
,	KUB23CSE131							3356
EXP	PERIMENT	EE137 X1823C	.823C5E1331ED	31 KUB23 C5E1	CSE131 KUB23	UB23C5E131	131 KUB23CST	33C5E137 WBV
5	ASSESTANCE SE	30,4	S.	E),	35	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	SE YUB	E \$ 13
M.	AGIC STRING	2 CSEN	T/1853	2/3/1	-13°C5*	T/BC	SENS	1823°
54.	escription	A KUBR?	(SK13)	.11823	31 408	CEL 3	F1853C	(13) KN 813°C
	Eva has a string S co characters in the stri	•	-				-	
_	Your task is to help E String. Return 0, if S		-	lue, representing	the minimum nu	mber of steps r	equired to form a	a Magic
,	Input Specification							V
(SEN3)	input1: A string S, co	ontaining lowerd	ase English lette	ers.				. 6
3	Output Specification	on:						1314
2	Return an integer value, representing the minimum number of steps required to form a Magic String. Return 0, if S is already a Magic String.							
3	Sample Input:							U823C54
	aaabbbccdddd							5
JB	Sample Output:							SEL 31
St. M. So	purce Codes	£131 £182 £131 £182 £131 £182 £131 £182 £131 £182 £131 £182 £131	13C5t131.	**************************************	E131 F18.	23 C5 K13.	31 K1823C.	5E131 L
T 18533	11823C5E1331 to	NB23C5E1	131 LIB23	335413	*11823c5*	[13] F19	B ²³ C5E1 ³³	13 Ma Barts Ma Barts Est
	F1855	3556137	17823CSF	131 TIB.	3C5E1331	*71813815	SEASON TO SEASON THE	38 15 Edit.
		*118533	SE131 (SE131)	1	,31 LUB2 . 31/3 KUBBE	**************************************	SEA BANG THE REPORT OF THE PARTY OF THE PART	SELIS THE BEEFER

1

```
def min_steps_to_magic_string(S):
              from collections import Counter
              \# Count the frequency of each character
              freq = Counter(S)
              # Find the maximum frequency
              max_freq = max(freq.values())
              # Total length of the string
              total_length = len(S)
              # Calculate the minimum steps needed
              min_steps = total_length - max_freq
              return min_steps
          # Sample Input
          input_string = "aaabbbccdddd"
          # Output the result
          print(min_steps_to_magic_string(input_string)) # Output: 8
1 / 5 Test Cases Passed | 20 %
```

RESULT