ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

$[48, 57] \rightarrow ['0', '9']$

$[65, 90] \rightarrow ['A', 'Z']$

$[97, 122] \rightarrow ['a', 'z']$

* Arithmatic operation over character it convert it self into ASCII code. such as $(+, -, >, <, =, *)$

(*) $\boxed{ch - 'A' = ch - 'a'}$

ex: $'p' - 'A' = 'p' - 'a'$

$'p' = 'p' - 'a' + 'A'$

Large: $ch = ch - 'a' + 'A'$

small: $ch = ch - 'A' + 'a'$

You are given a string `title` consisting of one or more words separated by a single space, where each word consists of English letters. **Capitalize** the string by changing the capitalization of each word such that:

- If the length of the word is `1` or `2` letters, change all letters to lowercase.
- Otherwise, change the first letter to uppercase and the remaining letters to lowercase.

Return *the* ***capitalized*** `title`.

```
def capitalizeTitle(self, title: str) -> str:
    l = len(title)
    if l <= 2:
        return self.lower(title)

    words = title.split(" ")
    wordsLen = len(words)

    res = ""
    for i in range(wordsLen):
        word = words[i]
        if len(word) <= 2:
            res += self.lower(word)
        else:
            ch = word[0]
            if self.isLowerCase(ch):
                res += self.convertSmallCharacterLarge(ch)
            else:
                res += ch

            res += self.lower(word[1:len(word)])

        if i != wordsLen - 1:
            res += " "
```

Count > 1 2

a a a a b b b c c d e 7 7 7 p g g

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

d

~~12~~ 8

n

a → 4

b → 3

c → 2

d → 1

7 → 3

p → 1

g = 2

"a4b3c2de73pg2"

```
def removeDuplicatesFromSortedString(str):
    l = len(str)

    if l <= 1:
        return str

    res = str[0]
    i = 1
    while i < l:
        count = 1
        while i < l and str[i] == res[len(res) - 1]:
            i += 1
            count += 1

        if i < l:
            res += str[i]

        i += 1

    return res
```

"res = "abcde7pg"

* a b c d a b c d e 7

a = 2

b = 2

c = 2

d = 2

e = 1

7 = 1