Task 1:

An English text needs to be encrypted using the following encryption scheme. First, the spaces are removed from the text. Let L be the length of this text. Then, characters are written into a grid, whose rows and columns have the following constraints:
$\left\lfloor \sqrt{L} \right\rfloor \leq row \leq column \leq \left\lceil \sqrt{L} \right\rceil$, where $\lfloor x \rfloor$ floor function and $\lceil x \rceil$ ceil function.

Example:
$Message =$ `if man was meant to stay on the ground god would have given us roots`
After removing spaces, the string is 54, $\sqrt{54}$ is between 7 and 8, so it is written in the form of a grid with 7 rows and 8 columns.

i	f	m	a	n	w	a	s
m	e	a	n	t	t	o	s
t	a	y	o	n	t	h	e
g	r	o	u	n	d	g	o
d	w	o	u	l	d	h	a
v	e	g	i	v	e	n	u
s	r	o	o	t	s		

Table 1: Message in grid

Always ensure $rowsXcolumns \geq L$. For this message length, L, 54 (without spaces) and grid size 7 rows and 8 columns, $7X8 = 56 < L$.

$Encrypted\ Message =$ `imtgdvs fearwer mayoogo anouuio ntnnlvt wttddes aohghn sseoau`

Now, create a function to encode a message.

Function Description:
Complete the *encryption* function as described below:
encryption has parameter(s):

- string s: a string to encrypt
- string *encrypted_s*: an empty string to store the encrypted string

and returns:

- int *len_encrypted_s*: the length of *encrypted_s*

The *main* function and *encryption* would look like this:

```c
#include<stdio.h>
#include<string.h>
#define MAX_SIZE 82
#define MAX_SIZE_ENCRYPT_MESSAGE 92


int encryption(char* s,char* encrypted_s){
  /* code */
  return len_encrypted_s;
}

int main()
{
  char s[MAX_SIZE];
  char encrypted_s[MAX_SIZE_ENCRYPT_MESSAGE];
  scanf("%s",s);
  /* code */
  return 0;
}
```

Input Format:
One line of text, the string s,where $1 \leq length\_of\_s \leq 81$. String will only consist of only characters in the range ascii[a-z] and space.

Output Format:
Prints the encrypted string, *encrypted_s*.

Sample Input	**Sample Output**
haveaniceday	hae and via ecy
feedthedog	fto ehg ee dd