Given a string, ***s***, consisting of alphabets and digits, find the frequency of each digit in the given string.

Input Format

The first line contains a string, ***num*** which is the given number.

Constraints

1 ≤ len(num) ≤ 1000

All the elements of num are made of English alphabets and digits.

Output Format

Print ten space-separated integers in a single line denoting the frequency of each digit from ***0*** to ***9***.

Print ten space-separated integers in a single line denoting the frequency of each digit from ***0*** to ***9***.

Sample Input 0

a11472o5t6

Sample Output 0

0 2 1 0 1 1 1 1 0 0

Explanation 0

In the given string:

- ***1*** occurs two times.
- ***2, 4, 5, 6*** and ***7*** occur one time each.

The remaining digits ***0, 3, 8*** and ***9*** don't occur at all.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
```

```c
#include<stdio.h>
int main()
{
    char str[1000];
    scanf("%s",str);
    int hash[10]={0,0,0,0
    int temp;
    for(int i=0;str[...
    {
        temp=str[i]-'0';
        if(temp<=9&&temp>=
        {
            hash[temp]++;
        }
    }
    for(int i=0;i<=9;i++)
    {
        printf("%d ",hash
    }
    return 0;
}
```

	Input	Expec
✓	a11472o5t6	0 2 1
✓	lw4n88j12n1	0 2 1
✓	1v88886l256338ar0ekk	1 1 1

✓	a11472o5t6	0 2 1
✓	lw4n88j12n1	0 2 1
✓	1v88886l256338ar0ekk	1 1 1

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

Flag question

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

Note: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

Note: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

Input:

The first line consists of an integer *T* denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

Output:

For each test case, print the count in a new line.

Given a string, ***s***, consisting of alphabets and digits, find the frequency of each digit in the given string.

Input Format

The first line contains a string, ***num*** which is the given number.

Constraints

1 ≤ len(num) ≤ 1000

All the elements of num are made of English alphabets and digits.

Output Format

Print ten space-separated integers in a single line denoting the frequency of each digit from ***0*** to ***9***.

Sample Input 0

Sample Input 0

a11472o5t6

Sample Output 0

0 2 1 0 1 1 1 1 0 0

Explanation 0

In the given string:

- ***1*** occurs two times.
- ***2, 4, 5, 6*** and ***7*** occur one time each.

The remaining digits ***0, 3, 8*** and ***9*** don't occur at all.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char str[1000];
    scanf("%s",str);
    int hash[10]={0,0,0,0
    int temp;
    for(int i=0;str[i]!='
```

occur at all.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char str[1000];
    scanf("%s",str);
    int hash[10]={0,0,0,0
    int temp;
    for(int i=0;str[i]!='
    {
        temp=str[i]-'0';
        if(temp<=9&&temp>=
        {
            hash[temp]++;
        }
    }
    for(int i=0;i<=9;i++)
    {
        printf("%d ",hash
    }
    return 0;
}
```

	Input	Expec
✓	a11472o5t6	0 2 1
✓	lw4n88j12n1	0 2 1
✓	1v88886l256338ar0ekk	1 1 1

Answer: (penalty regime: 0 %)

```
1  lude<stdio.h>
2  main()
3  
4  char str[1000];
5  scanf("%s",str);
6  int hash[10]={0,0,0,0,0,0
7  int temp;
8  for(int i=0;str[i]!='\0';
9  {
10     temp=str[i]-'0';
11     if(temp<=9&&temp>=0)
12     {
13        hash[temp]++;
14     }
15 }
16 for(int i=0;i<=9;i++)
17 {
18     printf("%d ",hash[i])
19 }
20 return 0;
21 
```

	Input	Expec
✓	a11472o5t6	0 2 1
✓	lw4n88j12n1	0 2 1

	Input	Expec
✓	a11472o5t6	0 2 1
✓	lw4n88j12n1	0 2 1
✓	1v888861256338ar0ekk	1 1 1

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

Flag question

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

Note: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

Note: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

Input:

The first line consists of an integer *T* denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

Output:

For each test case, print the count in a new line.

Constraints:

1 ≤ T ≤ 10
1 ≤ length of string ≤ 10^5

SAMPLE INPUT

2

nBBZLaosnm

JHkIsnZtTL

SAMPLE OUTPUT

2

1

Explanation

In test case 1, a and o are the only vowels. So, count=2

Answer: (penalty regime: 0 %)

```
1  #include<stdio.h>
2  int main()
3  {
4      int t;
```

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        char str[100000];
        int count=0;
        scanf("%s",str);
        for(int i=0;str[i]
        {
            char c= str[i]
            if((c=='a') ||
            count++;

        }
        printf("%d\n",cour
    }
    return 0;
}
```

	Input	Expected	Got	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	

	Input	Expected	Got	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

Flag question

Given a sentence, ***s***, print each word of the sentence in a new line.

Input Format

The first and only line contains a sentence, ***s***.

Given a sentence, ***s***, print each word of the sentence in a new line.

Input Format

The first and only line contains a sentence, ***s***.

Constraints

1 ≤ len(s) ≤ 1000

Output Format

Print each word of the sentence in a new line.

Sample Input 0

This is C

Sample Output 0

This

is

C

Explanation 0

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char s[1000];
    scanf("%[^\n]s",s);
    for(int i=0;s[i]!='\0
    {
        if(s[i]!=' ')
        printf("%c",s[i])
        else
        printf("\n");
    }
    return 0;
}
```

Answer: (penalty regime: 0 %)

```
lude<stdio.h>
main()

char s[1000];
scanf("%[^\n]s",s);
for(int i=0;s[i]!='\0';i++
{
    if(s[i]!=' ')
    printf("%c",s[i]);
    else
    printf("\n");
}
return 0;

```

	Input	Expected
✓	This is C	This is C
✓	Learning C is fun	Learning

	Input	Expected

✓	`This is C`	`This`
		`is`
		`C`

✓	`Learning C is fun`	`Learning`
		`C`
		`is`
		`fun`

Passed all tests! ✓

Question **4**

Correct

Marked out of 1.00

Flag question

Input Format

You are given two strings, ***a*** and ***b***, separated by a new line. Each string will consist of lower case Latin characters ('a'-'z').

Output Format

Input Format

You are given two strings, ***a*** and ***b***, separated by a new line. Each string will consist of lower case Latin characters ('a'-'z').

Output Format

In the first line print two space-separated integers, representing the length of ***a*** and ***b*** respectively.

In the second line print the string produced by concatenating ***a*** and ***b*** (***a + b***).

In the third line print two strings separated by a space, ***a'*** and ***b'***. ***a'*** and ***b'*** are the same as ***a*** and ***b***, respectively, except that their first characters are swapped.

Sample Input

abcd

ef

Sample Output

4 2

abcdef

ebcd af

Explanation

a = "abcd"

b = "ef"

|a| = 4

|b| = 2

a + b = "abcdef"

a' = "ebcd"

b' = "af"

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char str1[10],str2[10
    int i=0,j=0;
    int count1=0,count2=0
```

```c
#include<stdio.h>
int main()
{
    char str1[10],str2[10
    int i=0,j=0;
    int count1=0,count2=0
    scanf("%s %s",str1,str
    while(str1[i]!='\0')
    {
        count1++;
        i++;
    }
    while(str2[j]!='\0')
    {
        count2++;
        j++;
    }
    printf("%d %d\n",coun
    printf("%s%s\n",str1,
    t=str1[0];
    str1[0]=str2[0];
    str2[0]=t;
    printf("%s %s",str1,s
    return 0;

}
```

	Input	Expected	Got	
✓	abcd	4 2	4 2	✓

```
21      str1[0]=str2[0];
22      str2[0]=t;
23      printf("%s %s",str1,s
24      return 0;
25
26  }
```

	Input	Expected	Got	
✓	abcd ef	4 2 abcdef ebcd af	4 2 abcdef ebcd af	✓

Passed all tests! ✓

Finish review

Quiz navigation

1 2 3 4

Show one page at a time

Finish review