Question **1**

Correct

Marked out of 3.00

Flag question

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input format :

Line 1 : Name(Single character)
Line 2 : Marks scored in the 3 tests separated by single space.

Output format :

First line of output prints the name of the student.
Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A
3 4 6

Sample Output 1 :

A
4

Sample Input 2 :

T
7 3 8

```c
#include<stdio.h>
int main()
{
    char n;
    int m1,m2,m3;
    scanf("%c",&n);
    scanf("%d %d %d",&m1,&m2,&m3);
    printf("%c\n",n);
    printf("%d",(m1+m2+m3)/3);
}
```

Some *C* data types, their format specifiers, and their most common bit widths are as follows:

- *Int ("%d"):* 32 Bit integer
- *Long ("%ld"):* 64 bit integer
- *Char ("%c"):* Character type
- *Float ("%f"):* 32 bit real value
- *Double ("%lf"):* 64 bit real value

Reading

To read a data type, use the following syntax:

scanf("`format_specifier`", &val)

For example, to read a *character* followed by a *double*:

```
char ch;
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

Printing

To print a data type, use the following syntax:

printf("`format_specifier`", val)

For example, to print a *character* followed by a *double*:

```
char ch = 'd';
double d = 234.432;
printf("%c %lf", ch, d);
```

Note: You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

Input Format

Input consists of the following space-separated values: *int*, *long*, *char*, *float*, and *double*, respectively.

Output Format

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

```
3 12345678912345 a 334.23 14049.30493
```

Sample Output

```
3
12345678912345
a
334.230
14049.304930000
```

Explanation

Print *int* ***3***,

followed by *long* ***12345678912345***,

followed by *char* ***a***,

followed by *float* ***334.23***,

followed by *double* ***14049.30493***.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    long l;
    char ch;
    float f;
    double d;
    scanf("%d %ld %c %f %lf",&n,&l,&ch,&f,&d);
    printf("%d\n",n);
    printf("%ld\n",l);
    printf("%c\n",ch);
    printf("%.3f\n",f);
    printf("%.9lf\n",d);
    return 0;
}
```

	Input	Expected	Got	
✓	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000	3 12345678912345 a 334.230 14049.304930000	✓

Passed all tests! ✓

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input

E

Output

69
D F

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    char e,b,c;
    scanf("%c",&e);
    b=e-1;
    c=e+1;
    printf("%d",e);
    printf("\n%c %c",b,c);
    return 0;
}
```

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests! ✓