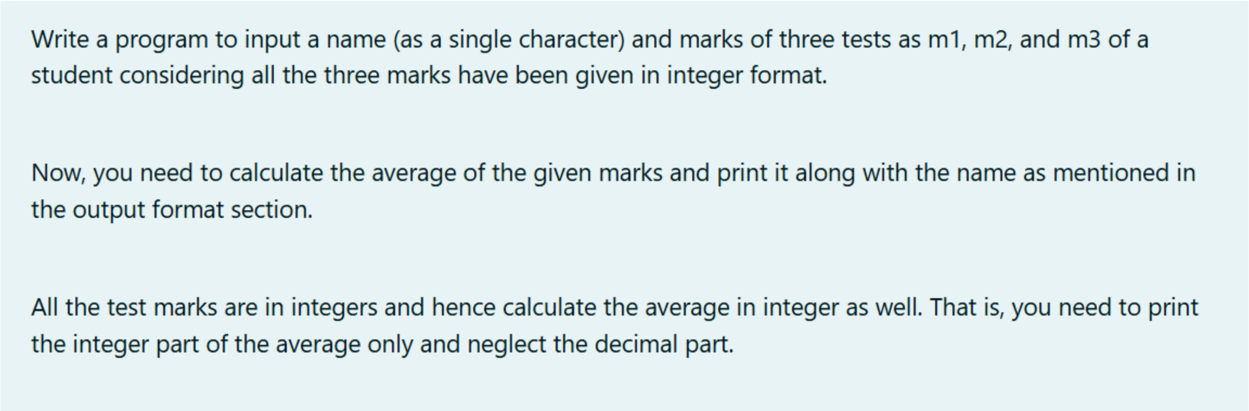
SANKARA GOMATHI.R

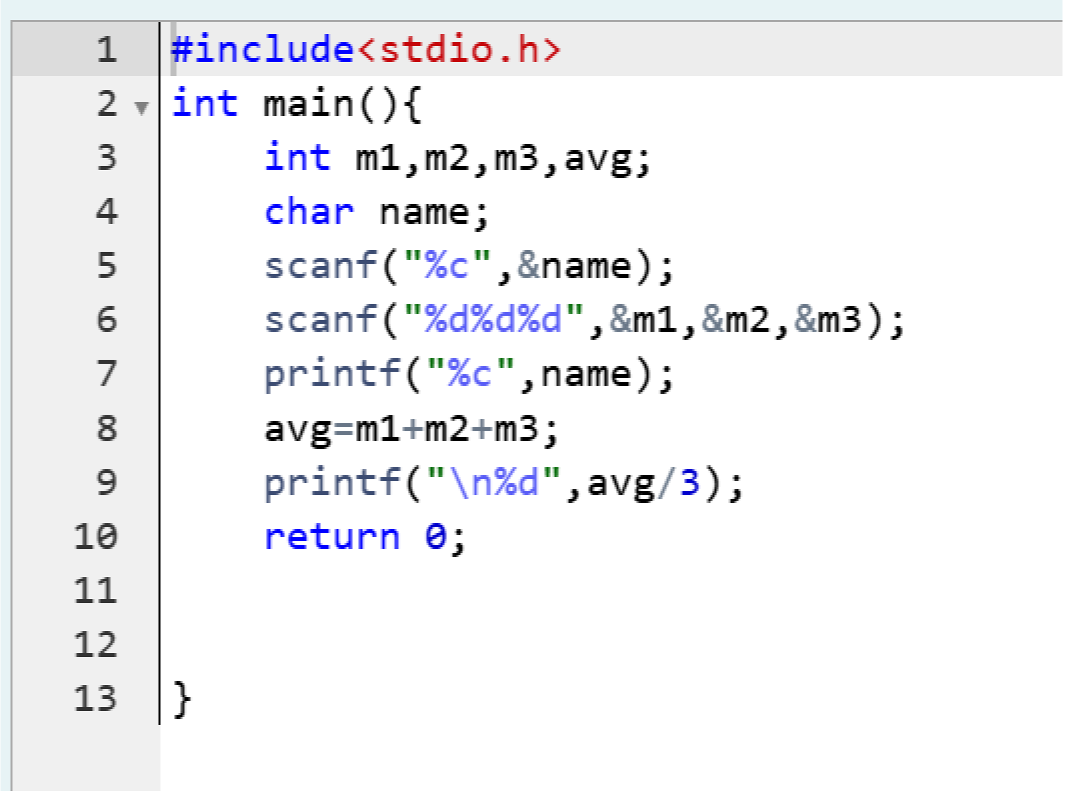
240701470



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.

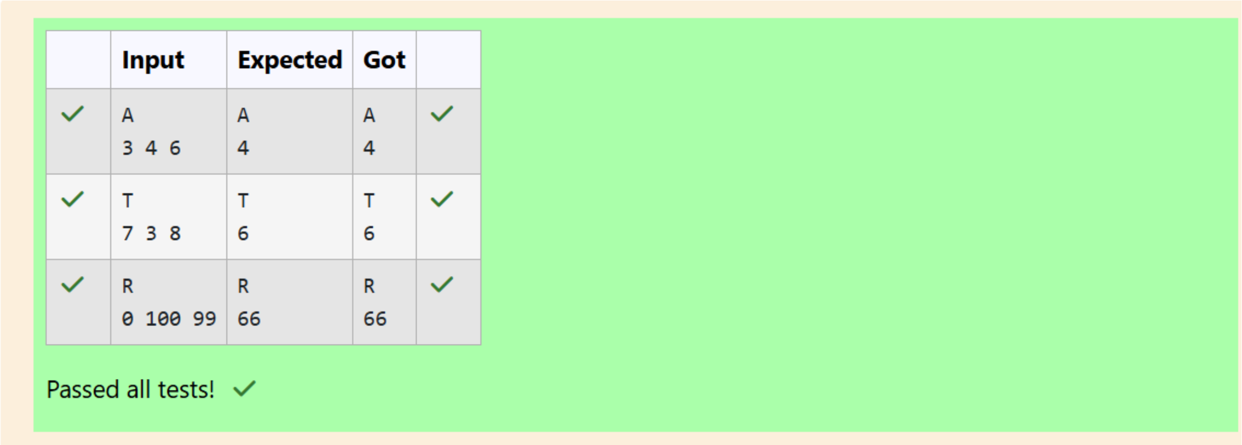
Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

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.



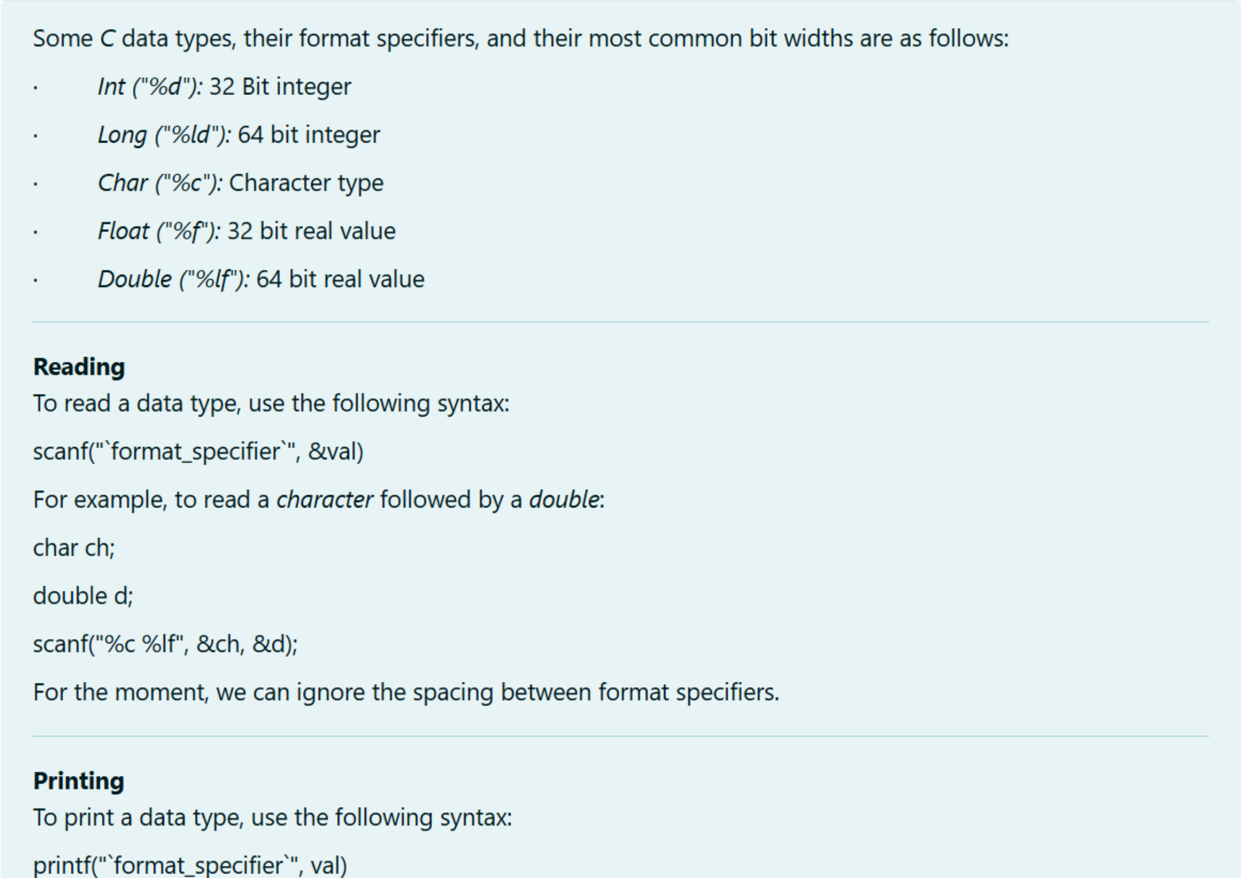
```c
#include<stdio.h>
int main(){
    int m1,m2,m3,avg;
    char name;
    scanf("%c",&name);
    scanf("%d%d%d",&m1,&m2,&m3);
    printf("%c",name);
    avg=m1+m2+m3;
    printf("\n%d",avg/3);
    return 0;


}
```



	Input	Expected	Got	
✓	A 3 4 6	A 4	A 4	✓
✓	T 7 3 8	T 6	T 6	✓
✓	R 0 100 99	R 66	R 66	✓

Passed all tests! ✓



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)

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)

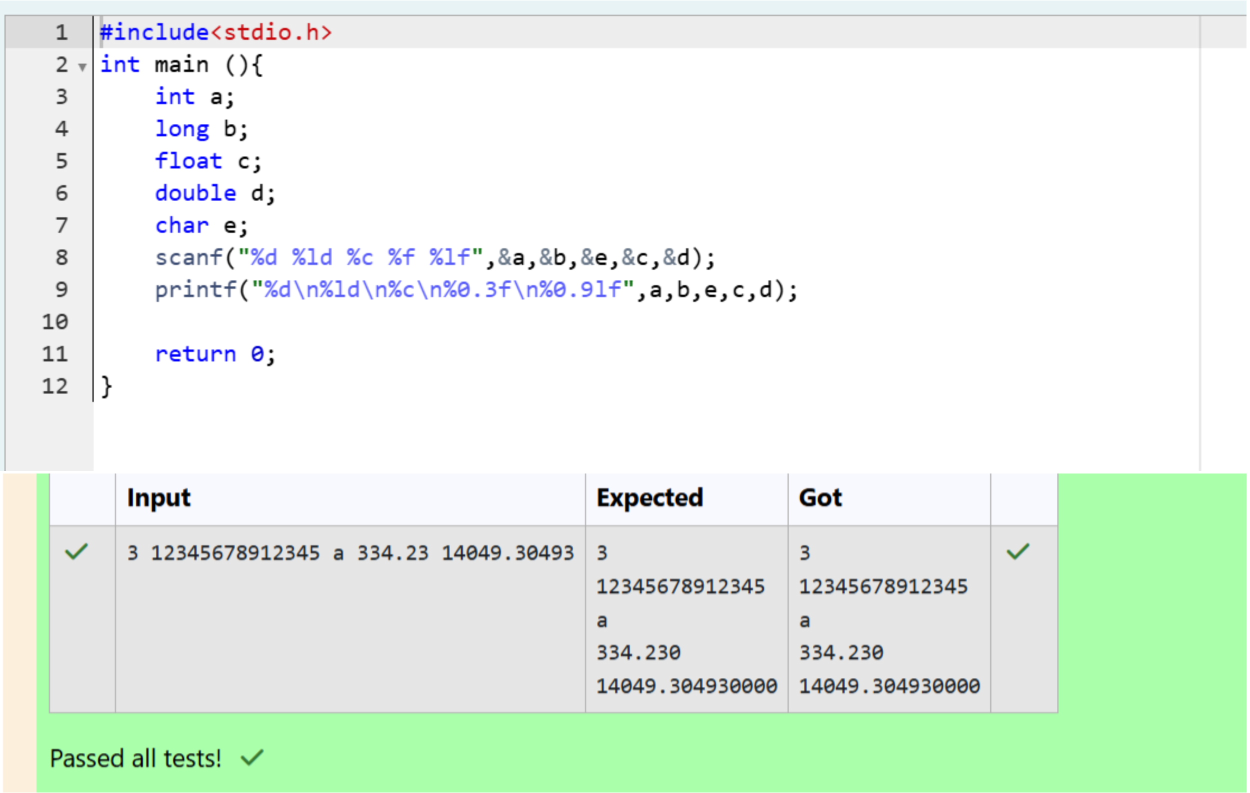
For example, to print a *character* followed by a *double*:

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*.



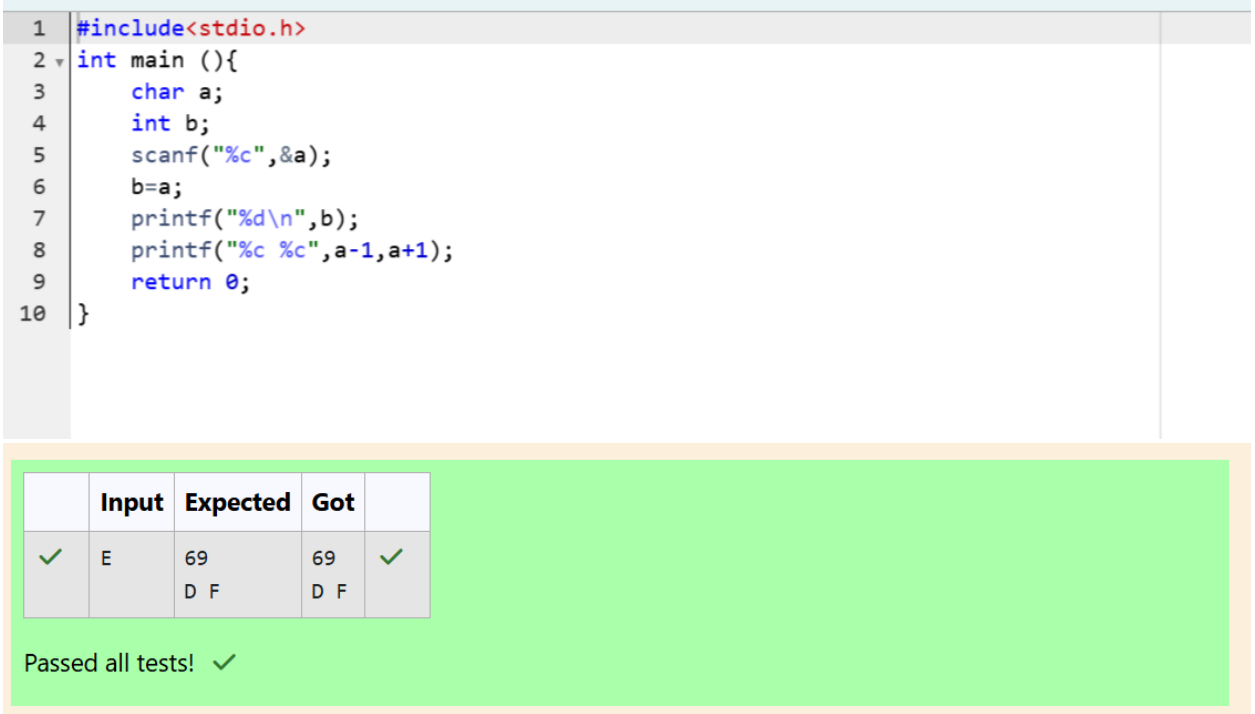
```c
#include<stdio.h>
int main (){
    int a;
    long b;
    float c;
    double d;
    char e;
    scanf("%d %ld %c %f %lf",&a,&b,&e,&c,&d);
    printf("%d\n%ld\n%c\n%0.3f\n%0.9lf",a,b,e,c,d);

    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.



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

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

Passed all tests! ✓