

Status	Finished
Started	Tuesday, 4 November 2025, 1:00 PM
Completed	Tuesday, 4 November 2025, 1:22 PM
Duration	21 mins 43 secs

Question **1**

Correct

A single line L with a set of space separated values indicating distance travelled and time taken is passed as the input. The program must calculate the average speed S (with precision upto 2 decimal places) and print S as the output.

Note: The distance and time taken will follow the format DISTANCE@TIMETAKEN. DISTANCE will be in kilometers and TIMETAKEN will be in hours.

Input Format:

The first line contains L.

Output Format:

The first line contains the average speed S.

Boundary Conditions:

Length of L will be from 3 to 100.

Example Input/Output 1:

Input:

60@2 120@3

Output:

36.00 kmph

Explanation:

Total distance = $60+120 = 180$ km.

Total time taken = $2+3 = 5$ hours.

Hence average speed = $180/5 = 36.00$ kmph

For example:

Input	Result
60@2 120@3	36.00 kmph

Answer: (penalty regime: 0 %)

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  int main(){
5      char i[101];
6      fgets(i,sizeof(i),stdin);
7      i[strcspn(i,"\n")]=0;
8      char*token;
9      double fd=0.0;
10     double ft=0.0;
11     token=strtok(i," ");
12     while(token!=NULL){
13         char*atsign=strchr(token,'@');
14         if(atsign!=NULL){
15             *atsign='\0';
16             double d=atof(token);
17             double t=atof(atsign+1);
18             fd+=d;
19             ft+=t;
20         }
21         token=strtok(NULL," ");
22     }
23     double s=0.0;
24     if(ft>0)
25         s=fd/ft;
26     printf("%.2lf kmph\n",s);
27     return 0;
28 }

```

	Input	Expected	Got	
✓	60@2 120@3	36.00 kmph	36.00 kmph	✓

Passed all tests! ✓

Question **2**

Correct

The program must accept two numbers X and Y and then print their HCF/GCD.

Input Format:

The first line denotes the value of X.

The second line denotes the value of Y.

Output Format:

The first line contains the HCF of X and Y.

Boundary Conditions:

$1 \leq X \leq 999999$

$1 \leq Y \leq 999999$

Example Input/Output 1:

Input:

30

40

Output:

10

Example Input/Output 2:

Input:

15

10

Output:

5

For example:

Input	Result
30 40	10

Answer: (penalty regime: 0 %)

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

```
2  int main() {  
3      int x,y,hcf;  
4      scanf("%d %d",&x,&y);  
5      while(x!=y){  
6          if(x>y)  
7              x=x-y;  
8          else  
9              y=y-x;  
10     }  
11     hcf=x;  
12     printf("%d\n",hcf);  
13     return 0;  
14 }
```

	Input	Expected	Got	
✓	30 40	10	10	✓

Passed all tests! ✓

Question **3**

Correct

A string S is passed as input. S will contain two integer values separated by one of these alphabets - A, S, M, D where

- A or a is for addition
- S or s is for subtraction
- M or m is for multiplication
- D or d is for division

The program must perform the necessary operation and print the result as the output. (Ignore any floating point values just print the integer result.)

Input Format:

The first line contains S.

Output Format:

The first line contains the resulting integer value.

Boundary Conditions:

Length of S is from 3 to 100.

Example Input/Output 1:

Input:

5A11

Output:

16

Explanation:

As the alphabet is A, 5 and 11 are added giving 16.

Example Input/Output 2:

Input:

120D6

Output:

20

Example Input/Output 3:

Input:

1405d10

Output:

140

For example:

Input	Result
5A11	16
120D6	20
1405d10	140

Answer: (penalty regime: 0 %)

```

1  #include<stdio.h>
2  #include<string.h>
3  #include<stdlib.h>
4  #include<ctype.h>
5  int main(){
6      char s[101];
7      scanf("%s",s);
8      int n1=0,n2=0;
9      char op=' ';
10     char*op_ptr=NULL;
11     for(int i=0;s[i]!='\0';i++){
12         if(isalpha(s[i])){
13             op=s[i];
14             op_ptr=&s[i];
15             break;
16         }
17     }
18     if(op_ptr!=NULL){
19         *op_ptr='\0';
20         n1=strtol(s,NULL,10);
21         n2=strtol(op_ptr+1,NULL,10);
22     }
23     int r=0;

```

```
24 switch(op){
25     case 'A':
26     case 'a':
27         r=n1+n2;
28         break;
29     case 'S':
30     case 's':
31         r=n1-n2;
32     case 'M':
33     case 'm':
34         r=n1*n2;
35         break;
36     case 'D':
37     case 'd':
38         if(n2!=0)
39         {
40             r=n1/n2;
41         }
42         break;
43     }
44     printf("%d\n",r);
45     return 0;
46 }
47 }
```

	Input	Expected	Got	
✓	5A11	16	16	✓
✓	120D6	20	20	✓
✓	1405d10	140	140	✓

Passed all tests! ✓