

<b>Status</b>	Finished
<b>Started</b>	Sunday, 2 November 2025, 5:48 PM
<b>Completed</b>	Sunday, 2 November 2025, 6:33 PM
<b>Duration</b>	44 mins 43 secs

**Question 1**

Correct

The name and mileage of certain cars is passed as the input. The format is CARNAME@MILEAGE and the input is as a single line, with each car information separated by a space. The program must print the car with the lowest mileage. (Assume no two cars will have the lowest mileage)

**Input Format:**

The first line contains the CARNAME@MILEAGE separated by a space.

**Output Format:**

The first line contains the name of the car with the lowest mileage.

**Boundary Conditions:**

The length of the input string is between 4 to 10000.

The length of the car name is from 1 to 50.

**Example Input/Output 1:**

Input:

Zantro@16.15 Zity@12.5 Gamry@9.8

Output:

Gamry

**For example:**

Input	Result
Zantro@16.15 Zity@12.5 Gamry@9.8	Gamry

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 #include<limits.h>
5 int main()
6 {

```

```
7     char input[10001];
8     fgets(input,sizeof(input),stdin);
9     char *token=strtok(input, " ");
10    char mincar[51]="";
11    double minmilage=1e9;
12    while(token!=NULL)
13    {
14        char carname[51];
15        double milage;
16        sscanf(token, "%[@]@%lf",carname,&milage);
17        if(milage<minmilage)
18        {
19            minmilage=milage;
20            strcpy(mincar,carname);
21        }
22        token=strtok(NULL, " ");
23    }
24    printf("%s",mincar);
25    return 0;
26 }
27
```

[]

	Input	Expected	Got	
✓	Zantro@16.15 Zity@12.5 Gamry@9.8	Gamry	Gamry	✓

Passed all tests! ✓

**Question 2**

Correct

A certain number of people attended a meeting which was to begin at 10:00 am on a given day. The arrival time in HH:MM format of those who attended the meeting is passed as the input in a single line, with each arrival time by a space. The program must print the count of people who came late (after 10:00 am) to the meeting.

**Input Format:**

The first line contains the arrival time separated by a space.

**Output Format:**

The first line contains the count of late comers.

**Boundary Conditions:**

The length of the input string is between 4 to 10000.

The time HH:MM will be in 24 hour format (HH is hours and MM is minutes).

**Example Input/Output 1:**

Input:

10:00 9:55 10:02 9:45 11:00

Output:

2

Explanation:

The 2 people were those who came at 10:02 and 11:00

**For example:**

Input	Result
10:00 9:55 10:02 9:45 11:00	2

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char input[10001];
6     fgets(input,sizeof(input),stdin);
7     int count=0;
8     char *token=strtok(input," ");
9     while(token!=NULL)
10    {
11        int hour,minute;
12        sscanf(token,"%d:%d",&hour,&minute);
13        if(hour>10||(hour==10&&minute>0))
14        {
15            count++;
16        }
17        token=strtok(NULL," ");
18    }printf("%d",count);
19
20 }
```



	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	10:00 9:55 10:02 9:45 11:00	2	2	✓

Passed all tests! ✓

**Question 3**

Correct

A single line consisting of a set of integers, each separated by space is passed as input to the program. The program must print the sum of all the integers present.

**Input Format:**

The first line contains the integer values (Each separated by a space)

**Output Format:**

The first line contains the sum of all the integers.

**Boundary Conditions:**

The length of the input string is between 3 to 10000

The value of the integer values will be from -99999 to 99999

**Example Input/Output 1:**

Input:

100 -99 98 5

Output:

104

**Example Input/Output 2:**

Input:

100 200 -300 500 -450 -50

Output:

0

**For example:**

Input	Result
100 -99 98 5	104
100 200 -300 500 -450 -50	0

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 int main()
5 {
6     char input[10001];
7     long long sum=0;
8     char *token;
9     fgets(input,sizeof(input),stdin);
10    token=strtok(input, " ");
11    while(token!= NULL)
12    {
13        sum +=atoi(token);
14        token=strtok(NULL, " ");
15    }
16    printf("%lld",sum);
17    return 0;
18 }
```



	Input	Expected	Got	
✓	100 -99 98 5	104	104	✓
✓	100 200 -300 500 -450 -50	0	0	✓

Passed all tests! ✓