

<b>Status</b>	Finished
<b>Started</b>	Monday, 3 November 2025, 3:48 PM
<b>Completed</b>	Monday, 3 November 2025, 4:05 PM
<b>Duration</b>	17 mins 9 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 int main()
5 {
6     char input[10000];
```

```

7 | fgets(input, sizeof(input), stdin);
8 | char* token = strtok(input, " ");
9 | char car[100], minCar[100];
10 | float mileage, minMileage = 999999.0;
11 | while(token != NULL){
12 |     sscanf(token, "%[^@]@%f", car, &mileage);
13 |     if(mileage < minMileage){
14 |         minMileage = mileage;
15 |         strcpy(minCar, car);
16 |     }
17 |     token = strtok(NULL, " ");
18 | }
19 | printf("%s\n", minCar);
20 | return 0;
21 | }

```

	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      char input[10000];
5      fgets(input,sizeof(input),stdin);
6      char*token;
7      int hour,min;
8      int count=0;
9      token=strtok(input," ");
10 do{
11     sscanf(token,"%d:%d",&hour,&min);
12     if(hour>10||(hour==10&&min>0)){
13         count++;
14     }
15     token=strtok(NULL," ");
16 }while(token!=NULL);
17 printf("%d\n",count);
18 return 0;
19 }
20

```



	Input	Expected	Got	
✓	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<string.h>
3  #include<stdlib.h>
4  int main(){
5      char input[10000];
6      fgets(input,sizeof(input),stdin);
7      char*token;
8      int num,sum=0;
9      token= strtok(input," ");
10 do{
11     if(token==NULL)
12         break;
13     num=atoi(token);
14     sum+=num;
15     token=strtok(NULL," ");
16
17 }while(token!=NULL);
18 printf("%d\n",sum);
19 return 0;
20 }
```



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

Passed all tests! ✓