

<b>Status</b>	Finished
<b>Started</b>	Sunday, 9 November 2025, 4:33 PM
<b>Completed</b>	Sunday, 9 November 2025, 4:59 PM
<b>Duration</b>	25 mins 44 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
9     char *token;
10    char lowestCar[100];

```

```
11     double lowestMileage = 9999999.0;
12
13     token = strtok(input, " ");
14     while(token != NULL){
15         char carname[100];
16         double mileage;
17
18         char *atsign = strchr(token, '@');
19         if(atsign != NULL) {
20
21             *atsign = '\0';
22             strcpy(carname, token);
23             mileage = atof(atsign + 1);
24
25         if(mileage < lowestMileage){
26             lowestMileage = mileage;
27             strcpy(lowestCar, carname);
28         }
29     }
30     token = strtok(NULL, " ");
31 }
32 printf("%s",lowestCar);
33 return 0;
34 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 | #include<stdlib.h> |
|   | .....              |

```
4 int time_to_minutes(const cchar* time_str){  
5     int hours, minutes;  
6     sscanf(time_str, "%d:%d", &hours, &minutes);  
7     return hours * 60 + minutes;  
8 }  
9 int main(){  
10    char input[10001];  
11    fgets(input, sizeof(input), stdin);  
12    input[strcspn(input, "\n")] = 0;  
13    char* token;  
14    int late_comers = 0;  
15    int meeting_start_time_minutes = -1;  
16    token = strtok(input, " ");  
17    if(token !=NULL){  
18        meeting_start_time_minutes = time_to_minutes(token);  
19    }  
20    while((token = strtok(NULL, " "))!=NULL){  
21        int arrival_time_minutes = time_to_minutes(token);  
22        if(arrival_time_minutes > meeting_start_time_minutes){  
23            late_comers++;  
24        }  
25    }  
26    printf("%d\n", late_comers);  
27    return 0;  
28 }
```

	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

Input	Result
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     fgets(input,sizeof(input),stdin);
8     long long sum = 0;
9     char*p = input;
10    char*end;
11    while(*p){
12        long num = strtol(p, &end, 10);
13        sum +=num;
14        if(p == end){
15            break;
16        }
17        p = end;
18    }
19    printf("%lld\n",sum);
20    return 0;
21 }
```

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

Passed all tests! ✓