

| | |
|------------------|-----------------------------------|
| Status | Finished |
| Started | Monday, 3 November 2025, 10:07 AM |
| Completed | Monday, 3 November 2025, 10:27 AM |
| Duration | 19 mins 50 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  int main()
4  {
5      char s[10000],name[50],minName[50];
6      float m,min=1e9;
7      fgets(s,sizeof(s),stdin);
8      char *p=strtok(s," \n");
9      while(p!=NULL)
10 {
```

```
11     sscanf(p, "%[^@]@%f", name, &m);
12     if(m < min)
13     {
14         min = m;
15         strcpy(minName, name);
16     }
17     p = strtok(NULL, " \n");
18 }
19 printf("%s\n", minName);
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  ▾ | {  
5      char s[10000],*p;  
6      int h,m,count=0;  
7      fgets(s,sizeof(s),stdin);  
8      p=strtok(s," ");  
9      while(p!=NULL)  
10     {  
11         sscanf(p,"%d:%d",&h,&m);  
12         if(h>10||(h==10 && m>0))  
13             count++;  
14         p=strtok(NULL," ");  
15     }  
16     printf("%d",count);  
17     return 0;  
18 }
```

| | 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  int main()
3  {
4      int n,sum=0;
5      while(scanf("%d",&n)==1)
6          sum+=n;
7      printf("%d",sum);
8      return 0;
9  }
```

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

Passed all tests! ✓