

#### **CSES Problem Set**

# **Elevator Rides**

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

#### **Submission details**

Task:	<u>Elevator Rides</u>
Sender:	Rodry
Submission time:	2021-11-28 01:11:02
Language:	C++11
Status:	READY
Result:	ACCEPTED

## **Test results** ▲

test	verdict	time	
#1	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#2	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#3	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#4	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#5	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#6	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#7	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#8	ACCEPTED	0.15 s	<u>&gt;&gt;</u>
#9	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#10	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#11	ACCEPTED	0.15 s	<u>&gt;&gt;</u>
#12	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#13	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#14	ACCEPTED	0.14 s	<u>&gt;&gt;</u>

## **Dynamic Programming**

Money Sums Removal Game ----Two Sets II Increasing Subsequence Projects Elevator Rides **Counting Tilings Counting Numbers** 

#### **Your submissions**

2021-11-28 01:11:02



test	verdict	time	
#15	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#16	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#17	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#18	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#19	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#20	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#21	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#22	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#23	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#24	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#25	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#26	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#27	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#28	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#29	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#30	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#31	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#32	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#33	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#34	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#35	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#36	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#37	ACCEPTED	0.13 s	<u>&gt;&gt;</u>
#38	ACCEPTED	0.01 s	<u>&gt;&gt;</u>

## **Compiler report** ▲

#### Code A

```
1 #include<bits/stdc++.h>
 3 #define INF 9999
   using namespace std;
 7 int ElevatorRides(int n, int x, vector<int>& v){
 8
            pair<int, int> resp[1<<n];</pre>
 9
10
       resp[0] = \{1,0\};
11
12
       int idx = (1<<n);</pre>
13
14
       for(int i=1;i<idx;i++){</pre>
15
16
            resp[i].first = n+1;
17
            resp[i].second = 0;
18
19
                     for(int j=0;j<n;j++){</pre>
20
21
                              int idx2 = (1 << j);
22
23
                if(i & idx2){
24
25
                                      pair<int,int> aux= resp[i^idx2];
26
27
                     if(aux.second+v[j] <= x)</pre>
28
                         aux.second += v[j];
29
30
                     else{
31
                         aux.first++;
32
                         aux.second=v[j];
33
34
35
                     resp[i]=min(resp[i],aux);
36
37
38
39
40
41
42
        return resp[idx-1].first;
43 }
```

```
44
45 int main(){
46
47
            int n,x;
48
49
       cin>>n>>x;
       vector<int> vector(n);
50
51
52
       for(int i=0;i<n;i++)</pre>
53
            cin>>vector[i];
54
55
            cout<<ElevatorRides(n,x,vector)<<endl;</pre>
56
```

## Share code to others

## Test details ▲

#### Test 1

Verdict: ACCEPTED



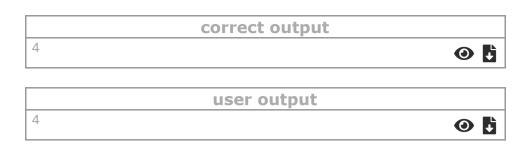


	user output	
3		<b>0</b>

#### Test 2

```
input

10 100
3 32 26 26 64 34 46 70 35 48
```



Verdict: ACCEPTED

	input	
10 100		
41 14 19 71 21 4	5 27 8 52 73	<b>O</b>

	correct output	
4		<b>0</b>

	user output	
4		<b>O</b>

## Test 4

									input	
10	100	9								
20	19	28	56	20	19	63	52	25	60	<b>O</b>

	correct output	
4		<b>0</b>

	user output
4	



Verdict: ACCEPTED

										input		
1	.0	100	9									
5	52	28	14	70	27	11	11	40	16	50	<b>•</b>	<b>↓</b>

	correct output	
4		<b>O</b>

user output	
4	<b>0</b>

## Test 6

Verdict: ACCEPTED

	input	
20 1000000000 609930576 743367699	654625611	<b>@</b>

correct output	
14	<b>O</b>

	user output	
14	<b>•</b>	<u>F</u>

## Test 7



correct output	
9	<b>©</b>

	user output	
9		<b>O</b>

Verdict: ACCEPTED

	input	
20 1000000000		
188194840 463651706	577460550	<b>O</b>

correct output	
8	<b>0</b>

	user output	
8		<b>O</b>

## Test 9

input	
20 1000000000 643426892 4189142 121707902 43	<b>0 b</b>

	correct output
6	



	user output	
6		<b>O</b>

Verdict: ACCEPTED

	input	
ı	20 1000000000	
	556514452 654521001 282817505	<b>0</b>

	correct output	
8		<b>② 5</b>

	user output	
8		<b>9</b>

## Test 11

	input	
20 10 2 5 10 4 4 5 4 3	9 2 10 2 7 8	<b>•</b>

	correct output	
10		<b>O</b>

	user output	
10		<b>0</b>

Verdict: ACCEPTED

input	
20 10	
9 2 10 2 6 9 6 5 1 6 2 9 3 3 4	<b>②</b>

	correct output	
11		<b>0</b>

	user output	
11		<b>0</b>

## Test 13

Verdict: ACCEPTED

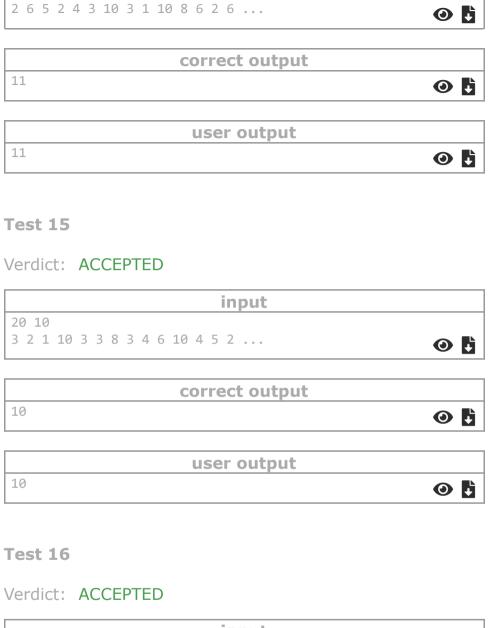


	correct output	
11		<b>0 b</b>

	user output	
11		<b>O</b>

#### Test 14

input
20 10





	correct output	
11		<b>0</b>



**Test 17** 

20 10	input	
	20 10 1 3 7 1 1 8 4 3 9 3 3 7 1 4 2	<b>0</b>

	correct output	
9	<b>0</b>	Ì

	user output	
9		<b>O</b>

## Test 18

input	
20 10 4 2 7 10 3 7 9 5 5 10 2 5 2 10	
7 2 7 10 3 7 3 3 10 2 3 2 10	<b>O</b>

	correct output	
11		<b>0</b>

	user output	
11	<b>⊙ €</b>	j

input	
20 10	
9 3 2 9 7 9 10 5 9 2 5 8 2 3 1	<b>•</b>

correct output	
12	<b>②</b>

user output	
12	<b>O</b>

## Test 20

Verdict: ACCEPTED

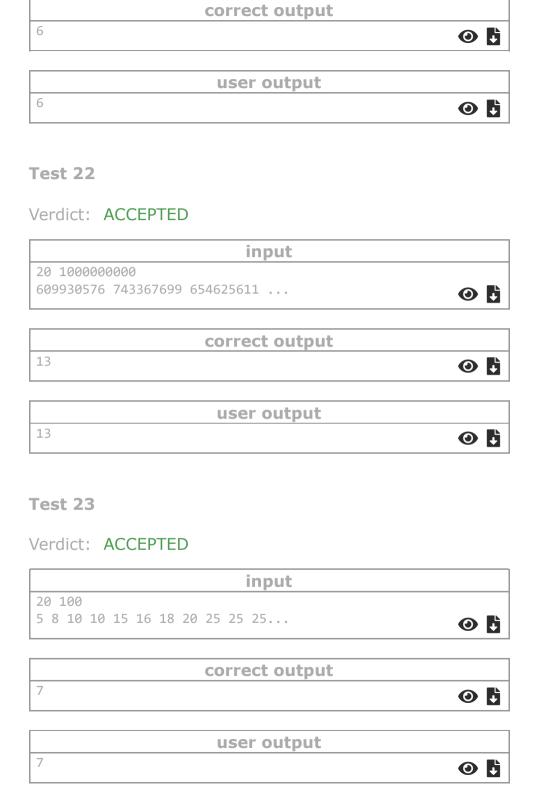
input	
20 10	
10 6 4 10 9 3 7 9 3 2 2 6 10 2	<b>O</b>

	correct output	
13		<b>0</b>

	user output	
13		<b>0</b>

#### Test 21

	input	
16 1000 589 17 199 306 495	559 14 269	<b>0 6</b>



Test 24

input	
20 10	_
	<b>O</b>

correct output	
2	<b>0</b>

	user output	
2		<b>②</b>

#### Test 25

Verdict: ACCEPTED

input	
4 10	
4 8 6 1	<b>O</b>

correct output	
2	<b>0</b>

	user output	
2		<b>Ø</b>

## Test 26



corı	ect output
7	<b>② L</b>

	user output	
7		<b>@</b>

Verdict: ACCEPTED

input	
20 100	
3 8 10 10 15 16 18 20 25 25 25	<b>O</b>

correct output	
7	<b>O</b>

	user output	
7		<b>O</b>

## Test 28

input	
20 9	
	<b>O</b>

correct output
6



Verdict: ACCEPTED

input	
20 100 41 42 43 44 45 46 47 48 49 59	<b>0 b</b>

correct output	
10	<b>O</b>

	user output
10	<b>② 5</b>

## Test 30

	input	
1 1		
1		<b>©</b>

	correct output	
1		<b>0</b>

	user output	
1		<b>0 L</b>

Test 31

	input	
2 2		
2 2		<b>②</b>

correct output	
2 <b>• • • • • • • • • • • • • • • • • • •</b>	ì

	user output	
2		<b>Ø</b>

#### Test 32

Verdict: ACCEPTED

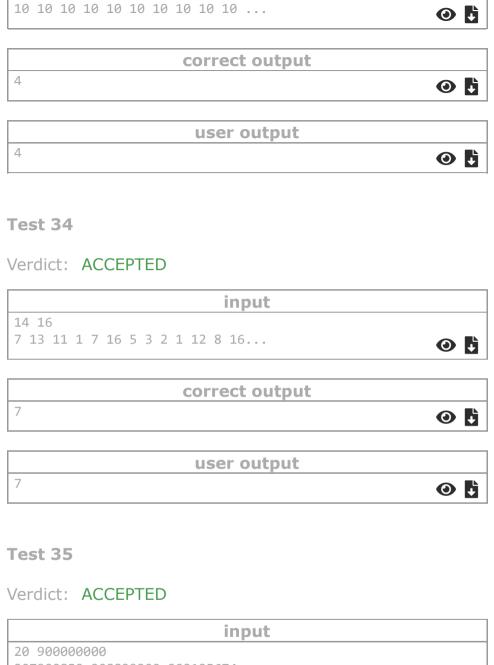


	correct output	
4		<b>0</b>

	user output	
4		<b>© B</b>

#### Test 33

input
20 69



input	
20 900000000	
207900850 208829300 203125674	<b>0</b>

	correct output	
6		<b>0</b>



Test 36

i	nput
20 1000000000 1000000000 100000	000
	<b>⊙ ↓</b>

	correct output	
20		<b>0 b</b>

	user output	
20		<b>O</b>

## Test 37

									in	put		
20	28	0										
56	56	56	56	56	40	40	40	40	40		<b>②</b>	<b>F</b>

	correct output	
3		<b>0</b>

	user output	
3		<b>0</b>

	input	
1 5		
2		<b>O</b>

	correct output	
1		<b>0</b>

user output		
1	<b>0</b>	