

A company plans to shift its infrastructure to the cloud. As a result, they will rent k computing cores per day for n days. The company is provided with m pricing plans for this duration. Each pricing plan has a $l[i]$ and $r[i]$ representing the days when it is available, $c[i]$ cores available each day, and it costs $p[i]$ to rent one core per day.

The company can rent an arbitrary number of cores on each day from each available plan, ranging from 0 to $c[i]$ cores. Given n , k , and $plans$ as a 2d array, the function returns the minimum cost to complete the process.

Example

Suppose $n = 5$, $k = 7$, and $plans = [[1, 3, 5, 2], [1, 4, 5, 3], [2, 5, 10, 1]]$

The optimal strategy is shown.

Day	Purchase cores/plan	Cost
1	5 from plan 1, 2 from plan 2	$5*2 + 2*3 = 16$
2	7 from plan 3	$7*1 = 7$
3	7 from plan 3	$7*1 = 7$
4	7 from plan 3	$7*1 = 7$
5	7 from plan 3	$7*1 = 7$

The sum of costs is $16+7+7+7+7 = 44$. Return 44.

Function Description

Complete the function `getMinCost` in the editor below.

getMinCost has the following parameter(s):

int n : the number of days to rent cores

int k : the number of cores to rent each day

int $plans[m][4]$: each $plan[i]$ contains $[l[i], r[i], c[i], p[i]]$, the start and end days, cores available, and price per core per day

Returns

int: the minimum cost to complete the process

Constraints

- $1 \leq n, m \leq 10^4$
- $1 \leq k \leq 500$
- $1 \leq plans[0], plans[1] \leq n$
- $1 \leq plans[2] \leq k$
- $1 \leq plans[3] \leq 10^9$
- It is guaranteed that for any available day there are at least k cores available.

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

STDIN

FUNCTION

```
-----  
4      →  n = 4  
4      →  k = 4  
4      →  plans[] size m = 4  
4      →  plans[][] size const = 4  
1 4 4 5 →  plans = [[1, 4, 4, 5], [1, 4, 2,  
5], [1, 2, 2, 1], [2, 3, 3, 2]]  
1 4 2 5  
1 2 2 1  
2 3 3 2
```

Sample Output

49

Explanation

Day	Purchase cores/plan	Cost
1	2 from plan 1, 2 from plan 3	$2*5 + 2*1 = 12$
2	2 from plan 3, 2 from plan 4	$2*1 + 2*2 = 6$
3	1 from plan 2, 3 from plan 4	$1*5 + 3*2 = 11$
4	7 from plan 3	$4*5 = 20$

▼ Sample Case 1

Sample Input For Custom Testing

STDIN

4 →

4 →

4 →

4 →

1 1 4 5 →

FUNCTION

n = 4

k = 4

plans[] size m = 4

plans[][] size const = 4

plans = [[1, 1, 4, 5]]

Sample Output

20

Explanation

On day 1, choose 4 cores from plan 1, $4 \times 5 = 20$.