



# STUDENT REPORT

## DETAILS

### Name

N NIKHIL

### Roll Number

3BR23CD063

## EXPERIMENT

### Title

DIWALI CONTEST

### Description

Max is planning to take part in a Diwali contest at a Diwali Party that will begin at 8 PM and will run until midnight (12 AM) i.e., for 4 hours. He also needs to travel to the party venue within this time which takes him **P** minutes. The contest comprises of **N** problems that are arranged in order of difficulty, with problem 1 being the simplest and problem N being the most difficult. Max is aware that he will require  $5*i$  minutes to solve the  $i^{th}$  problem.

Your task is help Max find and return an integer value, representing the number of problems Max can solve and reach the party venue within the given time frame of 4 hours.

Note: Max will leave his home at exactly 8 PM to reach the party venue.

### Input Format:

input1: An integer value N, representing the total number of problems.

input2: An integer value P, Representing the time to travel in minutes from his home to the party venue.

### Example:

#### Input:

6

180

#### Output:

4

### Explanation:

The amount of time left to solve the problems is  $4*60-180=60$  mins.

1st Problem - 5 mins, Time left =  $60-5=55$  mins

2nd Problem - 10 mins, Time left =  $55-10=45$  mins

3rd Problem - 15 mins, Time left =  $45-15=30$  mins

4th Problem - 20 mins, Time left =  $30-20=10$  mins

5th Problem - 25 mins

So he can solve only 4 problems as he is not left with 25 mins to complete 5th problem.

#### Source Code:

```
def max_problems_solved(N, P):  
    # Total available time for solving problems (240 minutes minus travel time)  
    remaining_time = 240 - P  
  
    # Initialize counters for time and problems solved  
    time_spent = 0  
    count = 0  
  
    # Iterate over problems from 1 to N  
    for i in range(1, N + 1):  
        # Time to solve the ith problem  
        time_to_solve = 5 * i  
  
        # Check if there's enough time left to solve this problem  
        if time_spent + time_to_solve > remaining_time:  
            break # Max can't solve more problems  
  
        # Update the time spent and count of problems solved  
        time_spent += time_to_solve  
        count += 1  
  
    return count  
N=int(input())  
P=int(input())  
result=max_problems_solved(N,P)  
print(result)
```

#### RESULT

5 / 5 Test Cases Passed | 100 %