



### Problem Solving (C54)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

## Gammal Tech's Smart Office System

**Background:** Gammal Tech, a trailblazer in the software industry, renowned for its innovative solutions and state-of-the-art office facilities, is developing a new technology to enhance its smart office systems. Their latest project aims to simplify office resource management, ensuring optimal utilization and comfort for their employees.

**Problem Description:** Your task is to help Gammal Tech develop a component of their smart office system. This component will manage the allocation of meeting rooms based on the number of participants and the duration of the meetings.

Each meeting room in Gammal Tech's office can accommodate a certain number of participants. The system should take as input the number of participants and the duration of the meeting and output the smallest meeting room number that can accommodate all the participants.

The system should also consider the duration of the meeting to optimize room usage. If a meeting is short (less than or equal to 30 minutes), a smaller room can be allocated,



even if it's a bit crowded. For longer meetings, comfort is prioritized, and larger rooms should be chosen.

### Input Format:

- The first line contains an integer  $T$ , the number of test cases.
- For each test case:
  - The first line contains two integers  $N$  and  $D$ , where  $N$  is the number of participants and  $D$  is the duration of the meeting in minutes.

### Output Format:

- For each test case, output a single line containing the room number that should be allocated. Assume the rooms are numbered starting from 1.

### Constraints:

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $1 \leq D \leq 180$

### Sample Input:

```
3
10 30
5 60
15 45
```

### Sample Output:

```
2
1
3
```



Explanation:

- In the first test case, there are 10 participants for a 30-minute meeting, so a medium-sized room (room 2) is allocated.
- In the second test case, there are 5 participants for a 60-minute meeting, requiring a comfortable space (room 1).
- In the third test case, 15 participants for a 45-minute meeting require a larger room (room 3).

للتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق

## C Programming Solution:

```
#include <stdio.h>

// Function to determine the room number
int findRoom(int participants, int duration) {
    if (duration <= 30) {
        return participants <= 10 ? 1 : 2;
    } else {
        if (participants <= 5) {
            return 1;
        } else if (participants <= 15) {
            return 2;
        } else {
            return 3;
        }
    }
}

int main() {
    int T, N, D;
    scanf("%d", &T);
    while (T--) {
        scanf("%d %d", &N, &D);
        printf("%d\n", findRoom(N, D));
    }
    return 0;
}
```