☆ The Perfect Team

The School of Languages and Science teaches five subjects: *Physics*, *Chemistry*, *Math*, *Botany*, and *Zoology*. Each student is skilled in one subject. The skills of the students are described by string of named *skills* that consists of the letters *p*, *c*, *m*, *b*, and *z* only. Each character describes the skill of a student as follows:

- *p* → *Physics*.
- *c* → *Chemistry*.
- *m* → *Math*.
- *b* → *Botany*.
- *z* → *Zoology*.

Your task is to determine the total number of different teams satisfying the following constraints:

- A team consists of a group of exactly *five* students.
- Each student is skilled in a different subject.
- A student may only be on one team.

For instance, if the *skills* string is *pcmbzpcmbz* then there are two possible teams that can be formed at one time: *skills[0-4]* and *skills[5-9]* for example. It is not important to determine permutations as we will always be limited to two teams given *10* students.

Function Description

Complete the function *differentTeams* in the editor below. The function must return an integer value representing the number of teams that can be formed given the constraints.

differentTeams has the following parameter(s):

skills: a string where each position represents the skill of a student

Constraints

- $5 \leq n \leq 5 \times 10^5$
- *skills[i]* $\in$ *{p,c,m,b,z}*

☆ Consecutive Sum

Given a long integer, find the number of ways to represent it as a sum of two or more consecutive positive integers.

For example, consider the number 21. It can be expressed as the sums of [1, 2, 3, 4, 5, 6], [6, 7, 8] and [10, 11]. There are 3 ways to sum to 21 using consecutive positive integers.

Function Description

Complete the function *consecutive* in the editor below. The function must return an integer denoting the number of ways to represent *num* as a sum of two or more consecutive positive integers.

consecutive has the following parameter(s):

num: the integer to sum to

Constraints

- $1 \leq num \leq 10^{12}$

☆ Work Schedule

You just got a new job where you have to work exactly as many hours as you are told each week, working no more than a daily maximum number of hours per day. Some of the days, they tell you how many hours you will work. You get to choose the remainder of your schedule, within the limits.

A completed schedule consists of exactly *7* digits in the range *0* to *8* representing each day's hours to be worked. You will get a pattern string similar to the schedule, but with some of the digits replaced by a question mark, *?, (ascii 63 decimal)*. Given a maximum number of hours you can work in a day, replace the question marks with digits so that the sum of the scheduled hours is exactly the hours you must work in a week. Return a string array with all possible schedules you can create, ordered lexicographically.

For example, your partial schedule, *pattern = 08??840*, your required hours, *work_hours = 24*, and you can only work, at most, *day_hours = 4* hours per day during the two days in question. You have two days on which you must work *24 - 20 = 4* more hours for the week. All of your possible schedules are listed below:

```
0804840
0813840
0822840
0831840
0840840
```

Function Description

Complete the function *findSchedules* in the editor below. The function must return an array of strings that represents all possible valid schedules. The strings must be ordered lexicographically.

findSchedules has the following parameter(s):

work_hours: an integer that represents the hours you must work in the week
day_hours: an integer that represents the maximum hours you may work in a day
pattern: a string that represents the partially completed schedule

Constraints

- *1 ≤ work_hours ≤ 56*
- *1 ≤ day_hours ≤ 8*
- *| pattern | = 7*
- Each character of *pattern* $\in$ *{0, 1,...,8}*