

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

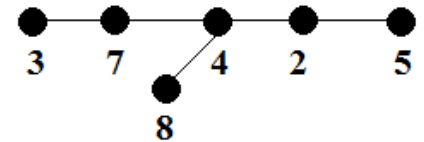
Русе, 7 – 9 юни 2019 г.

Група В, 9 – 10 клас

Задача В1. Туризм

Дадени са N града и $N-1$ пътища между тях, като има път между всеки два от тях. Някои от градовете са оценени от сайта turist.bg с коефициент K , в зависимост от забележителностите му. Група туристи от град A са решили да направят екскурзия до някой град B и да се върнат обратно по същия маршрут, спазвайки две изисквания. Първото – по маршрута AB сумата от коефициентите на посетените градове да е максимална (в тази сума не влиза коефициента на град A). Второто – искат всички градове от маршрута да се посетят по един път на отиване и по един път на връщане. На картинката при $A=3$ и $B=5$ е допустим маршрута 3-7-4-2-5, докато маршрутът 3-7-4-8-4-2-5 не е, защото град 4 се посещава два пъти в тази посока.

Напишете програма **turist**, която извежда намерената максимална сума от коефициенти от град A до някой град B , спазвайки дадените условия.



Вход

На първи ред е числото N - броя на всички градове. На следващите $N-1$ реда има по две числа P_i и Q_i , които означават, че има път между градове с номера P_i и Q_i .

На следващия ред е числото M - броя на градовете с оценка $K > 0$ и след него са M реда, също с по две числа G_i и K_i , които са съответно номера на града и коефициента му. На предпоследния ред е броя на тестовете T и на последния реда са числата A_i – номера на градове, от които ще започне екскурзията.

Изход

Изведете на един ред T на брой числа S_i , които означават, че за град A_i от входа, S_i е намерената максимална сума от коефициентите на посетените градове.

Ограничения: $1 \leq N, T \leq 200000$, $1 \leq M, P_i, Q_i, G_i \leq N$, $1 \leq K_i \leq 1000$.

Пример

Вход

9
3 6
3 4
6 1
1 7
4 2
4 5
2 8
2 9
5
3 3
2 1
5 2
9 4
7 3
3
3 2 8

Изход

5 6 7

Пояснение на примера:

От град с номер 3 по маршрута 3-4-2-9 по всеки път се минава по веднъж и сумата от коефициентите на върховете 4, 2 и 9 съответно е $0+1+4=5$.

Аналогично за град с номер 2 маршрутът е 2-4-3-6-1-7 и сума $0+3+0+0+3=6$, а за град с номер 8 маршрут: 8-2-4-3-6-1-7 със сума $1+0+3+0+0+3=7$.

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Русе, 7 – 9 юни 2019 г.

Група В, 9 – 10 клас

Задача В2. ХОН

Всяко изречение на планетата ХОН представлява съчетание от единствените три срички, произносими чрез специфичния говорен апарат на ХОН-айците. Ще споменем само, че всяко съчетание на трите срички представлява значещо изречение, предаващо многобройните нюанси на ХОН-айската душевност. Ще се условим също вместо трите йероглифа да използваме три от латинските букви, които също имат свойството да са идеално симетрични: Н, О и Х.

И сега най-важното: културата приема за голямо неуважение да се изписва името на планетата ХОН на дългите транспаранти, посрещащи гости-чужденци. Властите, разбира се, никога не го и правят. Проблемът е, че са се появили ХОН-айци-шовинисти, които през нощта замазват със спрей някои йероглифи от надписите, така, че остават четими само три: Х, О и Н, при това, представете си, в този ред! Скандал!

Ето един пример. Скромният надпис НХНОХХНО, който посреща чужденците на космодрум ООН, осъмва като НХНОХХНО! Филолози се изхитряват да заместят надписа с почти синонимичния му ОНХХООХО. „Да видим сега – доволно потриват пипала те – какво ще замажат!“ Това изглежда като перфектното решение: каквото и да замажат, няма да остане надпис ХОН. Какъв е ужасът, на сутринта: транспарантът с надписа се оказва обърнат с „главата“ нагоре (ОХООХХНО) и някои букви пак са замазани: ОХООХХНО!

Филолозите, разбира се, не се предават. След няколкочасово обсъждане те намират решението НХНХХХОО. Така шовинистите са поставени в невъзможност да приложат техниките си: нито в него, нито в обърнатия му обратно надпис ООХХХНХН могат да се замажат йероглифи така, че да остане четимо ХОН.

Като се знае, че на транспарантите могат да се напишат точно N йероглифа, пита се колко надписа могат да се направят, така че в никой от тях, нито прав, нито обърнат, да няма възможност така да се замажат йероглифи, че да останат точно три от тях, които да се четат ХОН. (За простота ще наречем такива изречения „хубави“.) Напишете програма **ХОН**, която намира остатъка на този брой при деление на въведено цяло положително число M .

Вход

От стандартния вход се въвежда един ред, в който има само две цели положителни числа N и M , разделени с интервал.

Изход

Програмата трябва да извежда на стандартния изход един ред, който съдържа само едно число: остатъка, който се получава при деление на броя различни „хубави изречения“, дълги по N йероглифа, на числото M .

Ограничения

Числото M не е по-голямо от 1 000 000 000 и в числото N има най-много 16 цифри.

В 20% от тестовите примери $N \leq 30$.

В 80% от тестовите примери $N \leq 100\,000\,000$.

Пример

Вход

4 11

Изход

8

Обяснение на изхода

„Хубавите“ надписи с дължина 4 йероглифа са 63 на брой: ОООО, ОООХ, ОООН, ООХН, ООХХ, ООХО, ООНХ, ООНН, ООНО, ОХНХ, ОХНН, ОХНО, ОХХН, ОХХХ, ОХХО, ОХОХ, ОХОО, ОНХХ, ОНХН, ОНХО, ОННХ, ОННН, ОННО, ОНОН, ОНОО, ХНХХ, ХНХН, ХНХО, ХННХ, ХННН, ХННО, ХНОО, ХХНХ, ХХНН, ХХНО, ХХХН, ХХХХ, ХХХО, ХХОХ, ХХОО, ХОХХ, ХОХО, ХООХ, ХООО, НХХХ, НХХН, НХХО, НХНХ, НХНН, НХНО, НХОО, ННХХ, ННХН, ННХО, НННХ, НННН, НННО, ННОН, ННОО, НОНО, НОНН, НООН и НООО. Остатъкът при деление на 63 на 11 е 8.

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

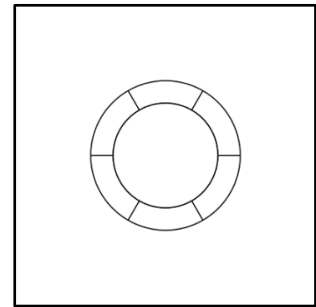
Русе, 7 – 9 юни 2019 г.

Група В, 9 – 10 клас

Задача В3. КОРИДОР

След като колегите на Криси във „Виртуалка Инк.“ няколко месеца неуморно я каниха, тя най-накрая отиде на ежеседмичното четвъртъчно *лимонадено* парти. Още преди да успее да си вземе *лимонада*, главата ѝ се замота и тя реши да си ходи. Докато неадекватно вървеше към изхода на сградата, тя погрешка влязла в една от стаите, в които „Виртуалка Инк.“ провежда своите прословути интервюта за стажанти, асистиран от техния патентован изкуствен интелект ГФнЖиДОС. Стаите за интервюта са тотално изолирани от външния свят и единственият вариант на Криси да излезе от там беше да реши теста, който се провежда в тази стая. От следващите 3 часа, ГФнЖиДОС прекара около 2 часа и 55 минути да залива Криси с ироничен монолог, а останалите около 5 минути – да ѝ обяснява какво точно представлява теста. В крайна сметка тя разбра следното:

Стаята, в която Криси се намира, представлява кръгъл коридор, разделен на N последователни сектора (виж **фиг. 1**) – в началото тя се намира в един тях. Във всеки от тези сектори има по една лампа. Всяка от лампите в началото е включена или изключена. Във всеки момент Криси вижда лампата в сектора, в който се намира (знае дали е включена или изключена), но не вижда нито една от другите лампи в коридора. Тя има право да извършва следните стъпки:



Фиг. 1. Коридорът при $N = 6$.

1. Да се премести в следващия по посока на часовниковата стрелка сектор.
2. Да се премести в следващия по посока обратна на часовниковата стрелка сектор.
3. Да включи или изключи лампата в сектора, в който се намира (ако лампата е включена може да я изключи, а ако е изключена – да я включи).
4. Да се опита да познае на колко е равно N .

Целта на Криси е да познае правилно стойността на N . Помогнете ѝ като реализирате алгоритъм, който тя да ползва. Понеже главата ѝ още се мотае, тя не иска да извършва твърде много стъпки и също така не може да помни много на брой неща – вашият алгоритъм ще е ограничен от страна на брой позволени стъпки и памет, която може да ползва между отделните стъпки (описано подробно по-долу).

Детайли по реализацията

Напишете функция `solve()` със следния прототип:

```
step solve(char cellValue);
```

Тя ще бъде извиквана многократно, симулирайки вашия алгоритъм стъпка по стъпка. Целта ѝ е да върне резултат, който описва каква е следващата стъпка при изпълнението на алгоритъма. При всяко извикване като аргумент ще ѝ бъде подавана стойност '0', ако лампата в сектора на коридора, в който се намирате, е включена, и '1' – ако е изключена. `solve` трябва да връща като резултат стойност от тип `step`, дефиниран по следния начин (**не трябва да го дефинирате сами**):

```
struct step {
    char action;
    int answer;
};
```

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Русе, 7 – 9 юни 2019 г.

Група В, 9 – 10 клас

1. За да укажете, че следващата Ви стъпка е преместване в следващия по посока на часовниковата стрелка сектор, трябва да върнете стойност буква 'l' на action (answer не се взема предвид).
2. За да укажете, че следващата Ви стъпка е преместване в следващия по посока обратна на часовниковата стрелка сектор, solve трябва да върне стойност от тип step, в която action е 'r' (answer не се взема предвид).
3. За да укажете, че следващата стъпка е включване или изключване на лампата в текущия сектор на коридора, solve трябва да върне стойност от тип step, в която action е 't' (answer не се взема предвид).
4. За да укажете, че следващата стъпка е опит за познаване на N, solve трябва да върне стойност от тип step, в която action е 'a', а answer е верния според Вас отговор. Имате право на един опит за познаване на броя сектори на даден коридор.
5. Счита се за грешка, ако solve върне резултат, в който action не е нито една от описаните по-горе стойности.

Тестовите ще бъдат групирани в групи от по **K** коридора. В рамките на един тест, вашето решение ще бъде симулирано върху всеки един от тях, като симулациите на отделните коридори ще бъдат преплетени една с друга – възможно е две последователни извиквания на solve да са за различни коридори (например може първите 2 извиквания да са за коридор 1, следващите 2 – за коридор 2, следващите 2 – отново за коридор 1 и т.н.). Всяка група носи точки, ако за всеки от коридорите ѝ е даден верен отговор.

Поради преплитането на симулациите на решението, е предоставен начин да имате памет, локална за всяка от тях. Всяка от симулациите разполага с памет в размер на 5 променливи от тип int (в началото на симулация, стойностите им са 0). Във функцията solve имате достъп до тази памет чрез функциите get_memory и set_memory (**не трябва да ги дефинирате сами**):

```
int get_memory (int index);  
void set_memory (int index, int value);
```

Извикването get_memory(i), връща стойността на i-тата (бройки от 0) променлива от паметта за текущата симулация. Извикването set_memory(i, x) присвоява стойност на i-тата променлива от паметта за текущата симулация стойност x. Последното ограничение за алгоритъма ви е, че при две негови симулации над еднакви коридори (с еднаква начална конфигурация на лампите и един и същ начален сектор), той трябва да работи по един и същ начин – трябва да прави едни и същи поредици от стъпки и поредици от извиквания на set_memory в двете симулации. Ако в рамките на един тест това не е вярно, тестът се счита за грешен. **Това значи, че всякакво съществено ползване на глобална памет би довело до грешно решение.**

Вие трябва да предадете към системата файл **corridor.cpp**, който съдържа функция solve(). Той може да съдържа и друг код и функции, необходими за работата на solve, но не трябва да съдържа главната функция main().

В началото си Вашият файл трябва да съдържа: **#include "corridor.h"**.

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Русе, 7 – 9 юни 2019 г.

Група В, 9 – 10 клас

Ограничения

$$1 \leq K \leq 5$$

$$2 \leq N \leq 200$$

Подзадачи

Подзадача	Точки	N	Максимален брой стъпки	Максимален брой извиквания на <i>set_memory</i>
1	20	≤ 50	500	1000
2	50	≤ 200	42000	84000
3	20	≤ 200	26000	52000
4	10	≤ 200	20000	40000

Локално тестване

За локално тестване са Ви предоставени файловете **corridor.h** и **Lgrader.cpp**. Сложете ги в същата папка, в която е Вашият файл **corridor.cpp** и компилирайте **Lgrader.cpp**. Така ще получите програма, с която ще проверите верността на функцията Ви. След стартиране на програмата, въведете на един ред низ, съставен от символите 0 и 1 (началният сектор е първият въведен символ). Програмата ще ви каже дали коректно сте намерили дължината, или не, както и дали не правите прекалено много стъпки или извиквания на *set_memory*.