Sim City

Termín odevzdání:	**24.12.2017 23:59:59**
Hodnocení:	**5.0000**
Max. hodnocení:	**5.0000** (bez bonusů)
Odevzdaná řešení:	2 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)
Nápovědy:	2 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat program, který bude fungovat jako pomůcka pro architekta ve hře Sim City.

Předpokládáme parcely umístěné v řadě podle silnice. Parcely mají čtvercový tvar a všechny mají stejnou velikost. Každá parcela může být použita jako místo pro dům (R - residential), místo pro komerci (C - commercial), místo pro průmysl (I - industrial) nebo jako park (P - park). Celkem je v řadě `n` parcel. Úkolem je určit, kolika různými způsoby lze nakombinovat použití parcel. Aby práce architekta nebyla tak jednoduchá, jsou pro použití parcel daná omezující pravidla. Je definovaný modul `m`, který omezuje:

- nejvýše `m` parcel v řadě může být bez parku,
- dvojice komerčních parcel musí mít mezi sebou nejméně `m` nekomerčních parcel,
- průmyslové parcely lze umístit nejméně 2 vedle sebe.

Vstupem programu je modul `m` následovaný požadavky na výpočet. Požadavky jsou dvou typů, každý požadavek je zadaný na zvláštním řádku:

- `list n prefix` - vypsat všechny možné platné kombinace použití `n` parcel při zadaném modulu `m`.
  Hodnota `prefix` nemusí být uvedena. Pokud je uvedena, udává předcházející parcely, na které se počítaných `n` dalších parcel napojuje. Napojení samozřejmě musí zachovat omezující pravidla. Parametr prefix je použit pouze v bonusových testech,
- `count n prefix` - vypsat počet možných platných kombinací použití `n` parcel při zadaném modulu `m`.
  Hodnota `prefix` má stejný význam jako u příkazu `list` (nemusí být uvedena, použita pouze v bonusových testech).

Zadávání požadavků na výpočty končí při dosažení konce vstupu.

Výstupem programu je buď počet existujících platných kombinací (dotaz typu `count`), nebo vypsaný seznam všech nalezených kombinací (dotaz typu `list`). Každá platná kombinace použití parcel je vypsaná na jedné řádce a je uzavřená do hranatých závorek. Kombinace začíná hodnotou `prefix` (pokud byla zadaná) a za ní následuje `n` parcel s vygenerovaným použitím. Použití parcely je vyjádřeno písmenem (P, I, R, C). Za seznamem následuje navíc počet nalezených kombinací. Pořadí nalezených kombinací na výstupu není dané, testovací prostředí si pořadí ve výpisu kombinací před porovnáním upraví.

Pokud je vstup neplatný, program to musí detekovat, zobrazit chybové hlášení a ukončit se. Chybové hlášení zobrazujte na standardní výstup (ne na chybový výstup). Za chybu považujte:

- není správně zadaný modul `m`: není celočíselný, je menší než 1 nebo větší než 10,
- není zadaný platný požadavek (platné jsou pouze `list` a `count`),
- není zadaná platná délka `n`: není celočíselná nebo je menší než 1,
- není zadaný platný `prefix`: prefix musí být tvořen kombinací písmen I, P, C a R. Testování prefixu je potřeba pouze u bonusových testů.

Program je testován v omezeném prostředí. Je omezena doba běhu a dostupná paměť. Časové i paměťové limity jsou nastavené tak, aby jimi prošla správná implementace postavená na naivním algoritmu. K dispozici jsou řádově desítky MiB paměti, které lze využít pro implementaci časově efektivního řešení (naivní řešení je paměťově nenáročné). Úloha má velkou časovou složitost. Naivní řešení má složitost exponenciální s velikostí `n`, pro zvládnutí bonusových testů je potřeba exponenciální řešení optimalizovat (první bonus) nebo zvolit zcela jiný postup (dynamické programování, druhý bonus).

Ukázka práce programu (zkráceno, kompletní výpis je v přiloženém archivu):

```
Modul:
3
Vypocty:
list 1
[P]
[R]
[C]
=> 3
list 2
[PP]
[PR]
```

[PC]
[II]
[RP]
[RR]
[RC]
[CP]
[CR]
=> 9
list 4
[PPPP]
[PPPR]
[PPPC]
[PPII]
[PPRP]

...

[CPRR]
[CIIP]
[CRPP]
[CRPR]
[CRRP]
=> 60
count 5
=> 159
count 12
=> 135946

Modul:
8
Vypocty:
list 4
[PPPP]
[PPPR]
[PPPC]
[PPII]
[PPRP]

...

[CRPP]
[CRPR]
[CRII]
[CRRP]
[CRRR]
=> 79
count 9
=> 12907
list abc
Nespravny vstup.

Modul:
38
Nespravny vstup.

Nápověda:

- Ukázkové běhy zachycují očekávané výpisy Vašeho programu (tučné písmo) a vstupy zadané uživatelem (základní písmo). Zvýraznění tučným písmem je použité pouze zde na stránce zadání, aby byl výpis lépe čitelný. Váš program má za úkol pouze zobrazit text bez zvýrazňování (bez HTML markupu).
- Bonusové testy testují větší hodnoty `n`, zadávají hlavně požadavky typu `count` (samotný výpis nalezených kombinací velmi zpomaluje).
- Počty nalezených kombinací mohou být velmi vysoké. V základních testech stačí pro reprezentaci datový typ `int`, pro bonusové testy je potřeba datový typ `long long int`.
- V bonusových testech se často zadávají požadavky s uvedeným prefixem. Podoba prefixu omezuje možnosti napojení následujících parcel. Tím se omezuje počet platných kombinací. Několik ukázek zadání s prefixem najdete v přiloženém

archivu v adresáři bonus. Pozor, prefix může být zadaný tak, že z něj nelze vygenerovat žádný platný výsledek (pak je počet řešení 0).