**Задача А4. Registers**

Климиразполага с неизправна регистърна машина. Във всеки регистър е запиано по едно цяло неотрицателно число. Машината поддържа програмен език със следния синтаксис:

Всеки непразен ред от програмата съдържа една от следните конструкции:

1. registers *<register\_1>* *<register\_2>* … *<register\_n>*

Декларация на използваните в програмата регистри. Тази команда трябва да присъства в програмата точно веднъж на първия непразен ред. Имената на регистрите могат да са произволни редици от ASCII символи различни от whitespace. Имената X, Y и Out са запазени и не могат да бъдат декларирани. **Можете да използвате най-много 12 регистъра.**

1. inc *<register>*

Машината опитва да увеличи стойността в регистъра с име *<register>* с 1. Тъй като е неизправна, има 50% шанс операцията да успее. При неуспех операцията няма ефект.

1. dec *<register>*

Машината опитва да намали стойността в регистъра с име *<register>* с 1. Тъй като е неизправна, има 50% шанс операцията да успее. При неуспех операцията няма ефект. Ако стойността в регистъра е била 0, то операцията няма ефект дори и при успех.

1. jeq *<register\_1> <register\_2> <label>*

Сравняват се стойностите на регистрите *<register\_1>* и *<register\_2>*. Ако те са равни то изпълнението продължава от мястото където е деклариран label-ът *<label>.* Забележете, че label-ът може да е деклариран както преди така и след текущата инструкция. Ако регистрите съдържат различни стойности то изпълнението продължава нормално към следващия ред.

1. *<label>*:

Декларира се label, който може да бъде използван от jeq инструкции. Имената на label-ите могат да са произволни редици от ASCII символи различни от whitespace.

1. #...

Всеки ред, чийто първи непразен символ е #, бива третиран като коментар и е еквивалентен на празен ред по време на изпълнение на програмата.

С изключение на скоци в резултат на jeq команди, редовете на програмата се изпълняват последователно. Програмата приключва, когато няма повече иинструкции за изпълнение, т.е. при стигане до края на файла. Програмата Ви може да ползва произволни whitspace символи (освен нови редове) с цел индентация или форматиране.

Помогнете на Клими да се научи да борави с машината като напишете програма registers.txt, която умножава две числа. За целта са Ви предоставени регистрите X, Y и Out. Първоначално в регистрите X и Y се съдържат съответно стойностите и , а във всички други регистри се съдържа 0. Вие трябва да напишете програма, която завършва със стойност в регистър Out. С изключение на това, че не трябва да декларирате тези три специални регистъра, те могат да бъдат ползвани по същия начин като всички останали регистри.

**Локално тестване**

С цел да тествате решението си, Ви е предоставен интерпретатор. На него му се задават числата и заедно с код на програма. След това той „компилира“ програмата, изпълнява я, и отпечатва върнатия резултат (или съобщение за грешка), както и броя изпълнени операции.

**Вход на интерпретатора**

От първия ред на входа се въвеждат две числа: и . От втория ред нататък се въвежда кода на програмата. Кодът на програмата се чете до достигане на край на файл.

**Ограничения**

**Подзадачи**

|  |  |  |
| --- | --- | --- |
| Подзадача | Допълнителни ограничения | Точки |
| 1 |  | 10 |
| 2 |  | 20 |
| 3 |  | 15 |
| 4 |  | 55 |

**Оценяване**

Дефинираме броя изпълнени операции като броя успешно изпълнени операции от тип inc, dec и jeq. Забележете, че неуспешните изпълнения на inc и dec **не** се броят.

За първите три подзадачи получавате пълните точки ако пресметнете вярното произведение за всеки тест с .

За четвърта подзадача дефинираме факторът . За получавате пълен брой точки. За получавате точки като частта е:

**Примерна програма**

|  |
| --- |
| # register with zero value  registers zero  # if X = Y jump to gotoInc  jeq X Y gotoInc  # terminate  jeq zero zero end  gotoInc:  inc Out  # label so we can jump to ending  end: |

**Обяснение на примерната програма**

Дадената примерна програма има следното поведение:

* За програмата винаги терминира с 0 в регистър Out.
* За програмата опитва да изпълни една команда inc. Съответно има точно 50% шанс да завърши с 0 в регистър Out и 50% шанс да завърши с 1.