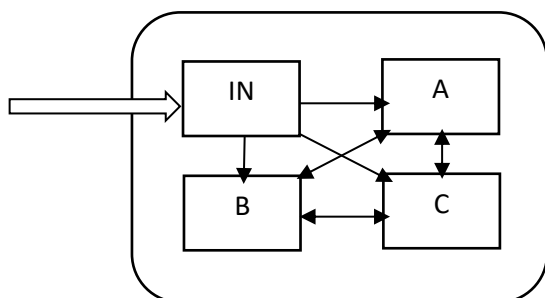


Задача 1. PF

Простата архитектура на нанопроцесора Perfect Function (съкратено PF) включва един входен регистър IN и три целочислени регистъра A, B и C. IN е свързан със стандартния вход, а резултатът от изчислителния процес се очаква в A. От стандартния вход в IN може да бъде въвеждано цяло неотрицателно число или един от символите +, -, * и =.

По-долу са описани елементарните операции, които PF може да извършва.



В таблицата са означени:

<Reg> - Всеки от четирите регистъра

<R>, <R₁>, <R₂> - Всеки от трите числови регистъра A, B или C

<number> - Цяло неотрицателно число

<const> - Някой от позволените символи или цяло неотрицателно число

(Можете да забележите, че в двуместните операции, с които се променя регистър, променящият се, е втори параметър.)

Код на операцията	Действие
ReadInt	От стандартния вход в IN се въвежда цяло неотрицателно число. Ако след изчистването на разделителите на стандартния вход има друго, възниква грешка.
ReadChar	От стандартния вход в IN се въвежда символ +, -, * или =. Ако след изчистването на разделителите на стандартния вход има друг символ или число, възниква грешка.
If <Reg> Is <const>	Начало на блок, който се изпълнява, ако съдържанието на Reg съвпада с константата const. Опит да се сравнят число и символ предизвиква грешка.
EndIf	Край на If-блока
Store <number> <R>	Съдържанието на R става равно на цялото неотрицателно число <number>
Copy IN <R>	Съдържанието на IN се копира в R. Ако в IN няма число, а символ, възниква грешка.
Copy <R ₁ > <R ₂ >	Съдържанието на R ₁ се копира в R ₂ .
Add <Reg> <R>	Съдържанието на Reg се добавя към това на R. Ако в Reg няма число, а символ, възниква грешка.
Sub <Reg> <R>	Съдържанието на Reg се изважда от това на R. Ако в Reg няма число, а символ, възниква грешка.
Mul <Reg> <R>	Съдържанието на R се умножава с това на Reg. Ако в Reg няма число, а символ, възниква грешка.
Repeat	Начало на блок за цикъл
Loop	Край на блок за цикъл
Stop	Край на изчислението. Резултатът е в A.

В началото на изчислителния процес всички регистри съдържат числото 0.

Задача

Напишете програма за процесора PF, която правилно изчислява аритметичен израз, съдържащ цели неотрицателни числа и аритметичните операции '+', '-' и '*'. Резултатът трябва да се получава в регистър A. Под „правилно“ разбираме да се спазва стандартно приетият приоритет на операциите: първо умноженията, след това събиране и изваждане от ляво надясно. Т. е., входът $2 + 2 * 2 =$ трябва да остави в A резултат 6, а не 8.

Входът към Вашата програма е коректен и ще изглежда по следния начин:

От стандартния вход към IN идва редица от редуващи се числа и по един символ. Тази редица започва с число. Всички числа са цели неотрицателни. Символът след всяко от тях е някой измежду '+', '-', '*' и '='. За разделител между членовете на редицата може да служи всяко непразно множество от интервали, табулатори или нови редове. За край на редицата определяме пристигането на символ '='. Символите '+', '-' и '*' изразяват общоприетите числови операции: съответно събиране, изваждане и умножение.

Изход

Това е задача от тип „output-only“. Изпратете на тестващата система текстов файл PF.txt, съдържащ написаната от Вас програма, решаваща поставената задача, всеки ред на който представлява правилна операция за процесора PF.

Ограничения

Файлът PF.txt не бива да надхвърля 255 програмни реда, както и да е повече от 64 KB. Резултатът от изчисленията на всяка подредица, започваща и завършваща с число, на входната редица е положително число, отрицателно число или нула, но не се записва с повече от 18 десетични цифри.

Оценяване

Всеки тест се оценява отделно. В 30% от тестовете се срещат само операции + и – или само операция *.

Интерпретатор

Предоставя Ви се интерпретатор на програми за нанопроцесора PF – файл PFI.cpp. Той чете от папката, в която е компилиран, файл PF.txt, който представлява програма за процесора PF, и очаква на стандартния вход данни за програмата. Ако се включи режим на трасиране (#define TRACE 1), инструкциите се изпълняват постъпково, след всяка се извежда на стандартния изход състоянието на регистрите и се изчаква клавиш <ENTER>. При възникване на грешка процесът прекъсва и на изхода за грешки се извежда съответно съобщение. Разбира се, можете да преправяте този интерпретатор, както Ви е удобно.

Пример.

Вход	Изход: Регистър А трябва да съдържа
12 - 0 * 12345678 * 5 + 3 * 6 - 43 =	-13

За по-бърза ориентация Ви показваме примерна програма за PF (идентичността също е примерна), която за коректно въведена редица от 4 члена: цяло неотрицателно число x, знак + или -, цяло неотрицателно число y и знак = пресмята израза $2^{x \pm y}$:

ReadInt	Stop	If B is 0	ReadChar
Copy IN B	EndIf	ReadChar	Stop
Store 1 A	ReadInt	If IN is +	EndIf
Copy A C	Sub IN A	ReadInt	Add C C
If B is 0	ReadChar	Add IN A	Mul C A
ReadChar	Stop	ReadChar	Store 1 C
If IN is +	EndIf	Stop	Loop
ReadInt	Add A A	EndIf	
Add IN A	Repeat	ReadInt	
ReadChar	Sub C B	Sub IN A	