**Анализ**

Задачата е от нестандартен тип – трябва да се пише на ограничен измислен език. Този език има единствено конструкции за смятане на изрази и използване на променливи. Няма условни изрази, цикли и т.н. Това, което трябва да се направи не е нищо особено, когато се разполага с условни конструкции. В случая задачата е интересна, защото трябва да се имитира такова поведение само с аритметични сметки.

Оценяването на задачата позволява да разглеждаме слепянето на числата и намирането на числото с минимална стойност като отделни подзадачи.

Лесни 10 точки, които могат да се вземат са в случая, когато *b* < 10. Това означава, че това число е едноцифрено и съответно за слепянето на числата е достатъчно да сметнем a\*10+b, а за втория отговор самото число *b* ни върши работа (то е числото с минимална стойност, като се вземат цифрите му).

Сега ще говорим как ще намерим слепеното число на *a* и *b* в общия случай, което носи 50 точки заради оценяването. Трябва по някакъв начин да разберем колко цифри има числото *b*, за да знаем с каква степен на 10 трябва да умножим *a* и да прибавим след това числото *b*, с което да получим стойността на слепеното число. Операцията, която е съществена в случая е целочисленото делене. Ако беше дробно, тогава нямаше да може да се реши задачата. Ще представим един примерен начин, по който може да подходим. Важно е да отбележим, че числото *b* е най-много трицифрено. Как може да разберем дали *b* е трицифрено? Ако сметнем b/100 можем да се ориентираме по резултата – ако е 0, значи не е трицифрено, иначе е трицифрено. Ще запазим в булева променлива дали е трицифрено (т.е. ще превърнем резултата b/100 в 0 или 1). Максималната стойност на b/100 е 9, това означава, че ако сметнем b/100+9, то ще получим 9, ако не е трицифрено и резултат от 10 до 18. Лесно може да се види, че ако разделим този резултат на 10, то ще го направим булев – той ще е 0, ако *b* не е трицифрено и 1, ако е. Така че сметката dig3=(b/100+9)/10 ще ни даде отговор за това дали *b* e трицифрено число или не. Аналогично, ако *b* не е трицифрено (b/10+9)/10 ще е булева стойност, определяща дали *b* е двуцифрено число или не. Какво да правим обаче, ако *b* вече е трицифрено? Можем да се възползваме, че вече знаем, че е трицифрено число като булев резултат. Ако сметнем dig2=(b/10+9)/10\*(1-dig3), то при трицифрено *b* тази сметка ще даде 0 и по-важното ще дава 1 само при двуцифрено число *b*. Отново като подходим аналогично, сметката dig1=(1-dig3)\*(1-dig2) ще ни даде булев резултат за това дали *b* е едноцифрено число. Вече остава лесната част – слепеното число ще е равно на a\*(1000\*dig3+100\*dig2+10\*dig1)+b.

Втората част от задачата е да намерим числото с най-малка стойност, използвайки цифрите на числото *b*. Ще направим едно наблюдение, което значително ще улесни разсъжденията и кода. Можем да разглеждаме тук *b* като трицифрено, като цифрите отпред, които евентуално може да ги няма, ще считаме за нули. Съответно това, което търсим ще е да сортираме тези три цифри в нарастващ ред, за да получим търсения резултат. Това отново не е толкова лека задача, защото нямаме условна конструкция. Нека сметнем в променливите digb3, digb2, digb1 цифрите на стотиците, десетиците и единиците на *b*. Ще приложим алгоритъма за сортиране на три променливи, който се учи още в E група – ще запишем след 2 сравнения минималната цифра в digb3, след което с още 1 сравнение ще сортираме digb2 и digb1. Остава да изясним как ще преценим от две цифри коя е минималната. Отново ще си послужим с целочисленото делене. Нека искаме да сравним digb3 и digb2. Ако сметнем div=digb2/digb3, ако div е 0, то digb2 е по-малко от digb3, иначе digb2 ≥ digb3. Удобно е да превърнем div в булев резултат. Ще подходим, както преди. Ясно е, че резултатът div ще е най-много 9 като стойност. Това означава, че сметката bool=(div+9)/10 ще е булев резултат, който е 0, ако div e 0 (съответно ако digb2 e по-малко от digb3) и 1 в противен случай. Така минималното число от digb3 и digb2 ще се дава от сметката digb3\*bool+digb2\*(1-bool). Аналогично можем да сметнем и максималното число от двете и в крайна сметка да присвоим на digb3 минималното, а на digb2 максималното. Тук обаче има един проблем, който по-наблюдателните трябва да са забелязали – може да делим на 0! За да избегнем това е достатъчно просто да прибавим 1-ца към двете цифри, т.е. div=(digb2+1)/(digb3+1). Описаната схема трябва да повторим като сравним digb3 с digb1, след което в digb3 ще е записана минималната цифра от трите и накрая да повторим същото за digb2 и digb1. Последно, търсеният резултат ще е равен на digb3\*100+digb2\*10+digb1.

*Автор: Илиян Йорданов*