

Задача 5.

```
1    br=0
2    br=`expr $br + $2`
3    set 3 5 7
4    shift
5    for j
6    do for var
7        do if test $br -lt $2
8            then br=`expr $br \* $2`
9                echo $br $var $j >> file
10           else br=`expr $br - $2`
11               echo $var $br $j >> file
12           fi
13       done
14       echo $*
15       break
16   done
17   read a1 a2
18   until cat file | grep $a1
19   do echo $#
20       grep $1 file
21       exit
22   done
23   echo $3
24   wc -l < file
25   echo FINAL
```

- На ред 1 създаваме променлива `br` и я инициализираме със стойност 0.
- На ред 2 актуализираме променливата `br` със стойност – старата стойност плюс стойността на втория аргумент на скрипта, който е 4. Командата `expr` пресмята стойността на израза който ѝ се подава и изписва резултата на стандартния изход, за това обграждаме целия израз с наклонени единични кавички, за да може да предадем тази стойност на променливата `br`. Новата стойност на `br` е 4.
- На ред 3 презаписваме стойностите на аргументите на скрипта. Аргументите вече са `$1=3`; `$2=5` и `$3=7`.
- На ред 4 изместваме аргументите, по подразбиране с 1 позиция. Командата `shift` освен, че прави изместването на входните аргументи се грижи и за актуализиране на броя им. Сега вече имаме два входни аргумента `$1=5` и `$2=7`.
- На ред 5 променливата `j` имплицитно цикли по входните аргументи. Командата е еквивалентна на `for j in "$@"`
- На ред 6 правим същото както на ред 5, с тази разлика, че променливата която цикли по входните аргументи вече не е `j`, а `var` и цикълът е вложен.
- На ред 7 проверяваме дали стойността на променливата `br` е строго по-малка от стойността на втория аргумент (`lt` = less than). $4 < 7$ (да), за това влизаме в `then` блока, където на ред 8 променливата `br` се актуализира на $4 * 7 = 28$. На ред 9 записваме във файл с име `file` съответно 28 5 5.
- На второто достигане до ред 7, `j` отново е 5, но `var` вече е 7. Тогава на ред 7 проверката няма да е изпълнена, тъй като 28 не е строго по-малко от 7 и скрипта влиза в `else` блока на `if-else` конструкцията, където `br` се актуализира на $28 - 7 = 21$ и във `file` се записва съответно 7 21 5.
- Нямаме повече аргументи по които да циклим във втория вложен цикъл и излизаме от него като на ред 14 принтираме на конзолата аргументите на скрипта, които са съответно 5 7.
- На ред 15 чупим външния цикъл и до този момент имаме принтиран един ред на конзолата и два реда във файла с име `file`.
- На ред 17 прочитаме от стандартния вход променливите `a1` и `a2` и им задаваме съответно стойностите 6 и 7.

- На ред 18 предаваме съдържанието на file чрез командата cat на командата grep, която проверява дали има 6-ца на някой от редовете. Няма ред, в който да има съвпадение и командата връща 1, което се третира като false. Тъй като until се изпълнява докато булевия израз в него не стане true, то ще се влезе в блока му и ще се принтира на ред 19 броя на параметрите на скрипта, който е 2.
- На ред 20 ще се потърси съвпадение във всеки ред на file с първия аргумент на скрипта, който е 5 и ще се върнат всички редове, т.е. ще се принтира файла file на конзолата. След това на 21 ред програмата ще приключи (exit).

Отговор:

```
5 7
2
28 5 5
7 21 5
```

