

### Задача 5.

```
1    br=1
2    br=`expr $br \* $2`
3    a=$3
4    set ab bc cd de
5    shift
6    while true
7    do echo $*
8        for j
9        do if test $# -lt $br
10            then br=`expr $br / 2`
11                echo $br $j >> file
12            else br=`expr $br + $a`
13                echo $br $j >> file
14            fi
15        done
16    break
17    done
18    read a1 a2
19    while cat file | grep $a1
20    do echo $a $a2
21        wc -l file
22        tail -5c file
23    exit
24    echo FIN
25    done
26    echo $a $a1
27    wc -c file
28    tail -2l file
```

- На ред 1 създаваме променлива с име br и я инициализираме със стойност 1.
- На ред 2 актуализираме променливата br като присвояваме старата ѝ стойност умножена по стойността на втория аргумент, която е 3. Т.е.  $br = br * 3 = 1 * 3 = 3$ .
- На ред 3 създаваме нова променлива с име a със стойност равна на стойността на третия аргумент на скрипта, т.е. 5.
- На ред 4 актуализираме аргументите на скрипта и те вече са 4 броя: ab, bc, cd и de.
- На ред 5 отместваме аргументите с един наляво и те вече са 3 броя като първият е bc.
- На ред 6 стартираме на пръв поглед безкраен цикъл.
- На ред 7 принтираме аргументите на скрипта като стринг (без кавички израза \$\* е еквивалентен на \$@).
- На ред 8 циклим имплицитно по аргументите на скрипта.
- За всеки един от аргументите на скрипта проверяваме дали броя на аргументите е строго по-малък от стойността на променливата \$br (която ще се актуализира спрямо истинността на предиката, който тества командата test).
- На първия test предиката ще е лъжа и ще влезем в else блок-а, а на втория и третия test ще влезем в then блок-а. Първо br ще се актуализира на  $\$br = \$br + \$a = 3 + 5 = 8$  и във файл с име "file" ще се запише реда "8 bc". След това br ще се актуализира на  $\$br = \$br / 2 = 8 / 2 = 4$  и във файла ще се допише на нов ред "4 cd" и накрая отново br ще се актуализира като отново се раздели на 2 и ще се запише трети ред във файла, който ще е "2 de".
- След като изциклим всички аргументи на скрипта на ред 15, на ред 16 ще последва команда break, която ще счупи безкрайния цикъл while.
- На ред 18 запамятаваме на аргументите a1 и a2 стойностите "c" и "d" (спрямо условието на задачата).
- На ред 19 подаваме съдържанието на файла "file" на командата grep с аргумент стойността на \$a1. Тази команда проверява ред по ред за съвпадения със стойността на \$a1, която е "c" и намира такива на първия и втория ред и ги принтира на конзолата. След като е намерила съвпадение, командата grep връща код на изпълнение 0, който

контраинтуитивно се интерпретира като true от оператора while и следователно влизаме в do блок-а.

- На ред 20 принтираме на конзолата "5 d".
- На ред 21 командата wc (word count) с аргумент -l (lines) принтира на конзолата броя на редове във файла "file" и името на файла.
- На ред 22 командата tail с аргумент -5с (-5 chars) и име на файл принтира последните 5 символа от файла "file". Това са (отзад напред) : символа за нов ред '\n', 'e', 'd', ' ', '2'.
- Командата exit на ред 23 прекратява изпълнението на скрипта.

Отговор:

```
bc cd de
8 bc
4 cd
5 d
      3 file
2 de
```

