

Laboratoria nr 4

Podprogramy, rekursja i struktury w C

Przed uczestnictwem w zajęciach student powinien:

- podstawowe wiadomości o rekordach (struktury, unie), tablice rekordów,
- definiowanie typów za pomocą deklaracji `typedef`,
- definiowanie funkcji o zmiennej liczbie parametrów,
- dynamiczny przydział pamięci,
- operacje dyskowe.

Materiały

- http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_3.PDF
- http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_4.PDF
- http://www.cs.put.poznan.pl/wcomplak/BFILES/C_W_5.PDF

Zadania

1. Znajdowanie największej wartości wyrazu ciągu Fibonacciego nie większej od danej wartości podanej z wejścia (rekurencyjnie).
2. Napisać funkcję potęgowania liczby naturalnej za pomocą mechanizmu wielokrotnego dodawania (rekurencyjnie).
3. Napisać funkcję wyznaczania silni liczby naturalnej n (rekurencyjnie).
4. Liczba automorficzna to liczba, która znajduje się na końcu swego kwadratu, np.: $5^2 = 25$, $25^2 = 625$. Napisz program znajdowania wszystkich liczb automorficznych w podanym przedziale domkniętym $\langle a, b \rangle$.
5. Napisz program obliczania sumy cyfr dziesiętnych podanej liczby n (rekurencyjnie).
6. Napisać program pozwalający na zmianę reprezentacji liczb (dziesiętną, binarną, ósemkową, szesnastkową).
7. Implementacja następujących struktur danych: stos, lista jednokierunkowa, dwukierunkowa, drzewo BST – dostępne powinny być operacje: dodawania, usuwania i przeszukiwania.
8. Napisz program scalania ciągów, który polega na łączeniu posortowanych ciągów w jeden ciąg posortowany.
9. Zliczyć ilość wystąpień zdefiniowanego znaku w pliku wejściowym.
10. Napisać kalkulator pozwalający na dodawanie, odejmowanie, mnożenie liczb szesnastkowych.
11. Sortowanie stogowe.
12. Wypisanie najdłuższego i najkrótszego wiersza pliku wejściowego.
13. Zaimplementuj następujące funkcje występujące w języku AWK: `gsub(r,s,t)`, `gensub(r,s,h,t)`, `index(s,t)`, `split(s,a,fs)`, `sub(r,s,t)`.
14. Załóżmy, że w pliku wejściowym znajdują się wiersze zawierające tylko pola numeryczne. Napisz program, którego zadaniem będzie wypisanie, wartości bezwzględnych dla wszystkich pól znajdujących się w pliku wejściowym.

15. Napisać programy, które przeprowadzą konwersję plików według poniższych formatów:

```
* Line 1      | 1. Line 1
** Line 2    |  1.1. Line 2
*** Line 3   |   1.1.1. Line 3
*** Line 4   |   1.1.2. Line 4
**** Line 5  |    1.1.2.1. Line 5
***** Line 6 |    1.1.2.1.1. Line 6
***** Line 7 |    1.1.2.1.2. Line 7
** Line 8    |  1.2. Line 8
* Line 9     | 2. Line 9
** Line 10   |  2.1. Line 10

* Line 1     | A. Line 1
** Line 2   |  1. Line 2
*** Line 3  |   a. Line 3
*** Line 4  |   b. Line 4
**** Line 5 |   (1) Line 5
***** Line 6 |   (a) Line 6
***** Line 7 |   (b) Line 7
** Line 8   |  2. Line 8
* Line 9    | B. Line 9
** Line 10  |  1. Line 10
```

16. Napisz program, którego zadaniem będzie wypisanie statystyk związanych z wartościami numerycznymi znajdującymi się w kolumnach. W skład statystyk dla każdej kolumny ma zostać wypisana suma wszystkich pól tej kolumny oraz średnia związana z daną kolumną.

Plik wejściowy

```
1 2 3 5 2
0 10 12 2 3 1
```

Plik wyjściowy

```
Suma: 1 12 15 7 5 1
Średnia: 0,5 6 7,5 3,5 2,5 1
```

17. Przyjmijmy, że słowo jest to niepusty ciąg znaków różnych od spacji, znaku tabulacji i nowej linii, (zatem ciąg $a+b*c$ jest jednym słowem). Napisać program obliczania liczby słów w pliku. Wiersze, w których pierwszym znakiem jest średnik nie są brane pod uwagę.

Plik wejściowy

```
jeden 2 !!!
; on on on on on on on on on
;on on on on on on on on
four ;; six
```

Plik wyjściowy

```
6
```

18. Plik wejściowy zawiera ciąg wierszy. W każdym wierszu znajdują się trzy liczby całkowite (o małych wartościach bezwzględnych), oddzielone od siebie dowolną liczbą spacji i znaków tabulacji. Napisać program obliczający dla każdego wiersza sumę liczb w nim zawartych i drukujący zestawienie tabelaryczne uzyskanych wyników w postaci: nagłówek 'SUMA' dla ostatniej kolumny, w kolejnych wierszach, wyrównane w kolumnach do lewej składowe sumy i wartość sumy, oddzielone od siebie znakami '|'.

Plik wejściowy

```
1 5 18
13 2 -5
```

Plik wyjściowy

```
SUMA
| 1 | 5 | 18 | 24
| 13 | 2 | -5 | 10
```

19. Napisać program pozwalający na obliczanie wartości wyrażeń zapisanych za pomocą odwrotnej notacji polskiej *ONP* (np.: $+(-(3,2),1)=2$).
20. Napisać program kompresujący dane za pomocą algorytmów *ByteRun*, *kody Huffmana*, *LZW*.