5 Dynamiczne struktury danych

> Niestety, wciąż zbyt wielu osobom wydaje się, że przetwarzanie list jest skomplikowane.
>
> D.E. Knuth, *Sztuka programowania*

Ten zestaw testuje umiejętność definiowania i wykorzystywania podstawowych struktur danych: realokowanych dynamicznie tablic, kolejek oraz drzew binarnych.

0x2C Zadanie ❀

Napisać program, który czyta z wejścia ciąg liczb całkowitych aż do napotkania zera, a następnie wypisuje je na wyjściu w odwrotnej kolejności.

0x2D Zadanie ★

Mamy daną listę przechowującą liczby całkowite „w paczkach”:

```
#define ROZMIAR_PACZKI 1024

typedef struct wezel_ {
    struct wezel_ *nastepny, *poprzedni;
    int paczka[ROZMIAR_PACZKI];
    int uzyte;
} wezel;

extern wezel *pierwszy, *ostatni;
```

Pole `uzyte` określa jaka cześć danej paczki jest faktycznie używana. Napisać funkcję sortującą *całą* listę.

0x2E Zadanie ❀

Mamy daną listę jednokierunkową

```
typedef struct wezel_ {
    struct wezel_ *nastepny;
    int wartosc;
} wezel;
```

o której zakładamy, że jest cykliczna. Napisać funkcję przyjmującą jako argument wskaźnik na element tej listy i zwracającą w wyniku jej rozmiar.

0x2F Zadanie

Mamy daną listę:

```
typedef struct wezel_ {
    struct wezel_ *nastepny;
    int wartosc;
} wezel;
wezel *pierwszy;
```

Napisać funkcję sortującą tę listę.

0x30 Zadanie

Mamy daną listę:

```
typedef struct {
    char   *tytul;
    char   *autor;
    double cena;
    int    ilosc;
} ksiazka;

typedef struct wezel_ {
    ksiazka dane;
    struct wezel_ *nastepny, *poprzedni;
} wezel;

extern wezel *pierwszy;
```

Pola `autor` są współdzielone pomiędzy poszczególne elementy listy. Napisać funkcję usuwającą z listy wszystkie pozycje o zerowej ilości oraz zwalniającą pamięć przeznaczoną na personalia autorów, których książek już nie ma na liście.

0x31 Zadanie

Stworzyć strukturę danych reprezentującą trójkąt Pascala. Wykorzystując ją napisać funkcję zwracającą wartość symbolu Newtona.

0x32 Zadanie

Tablicę `float drzewo[ROZMIAR]` możemy traktować jako drzewo binarne, w którym lewym/prawym potomkiem elementu `drzewo[i]` są elementy o indeksach odpowiednio: `2*i + 1` oraz `2*i + 2`. Napisać funkcje przechodzenia takiego drzewa w kolejności preorder, inorder i postorder.

0x33 Zadanie

Napisać funkcję, która przekształca posortowaną tablicę liczb zmiennoprzecinkowych w doskonale zbalansowane drzewo, którego każdy węzeł spełnia warunek:

$$\text{lewe poddrzewo} \leq \text{węzeł} \leq \text{prawe poddrzewo}.$$

0x34 ZADANIE

Mamy dane drzewo binarne:

```
typedef struct wezel_ {
    long double dane;
    struct wezel_ *lewy, *prawy;
} wezel;

wezel *korzen;
```

Napisać funkcję, która wylicza długość najkrótszej drogi od korzenia do liścia tego drzewa.

0x35 ZADANIE

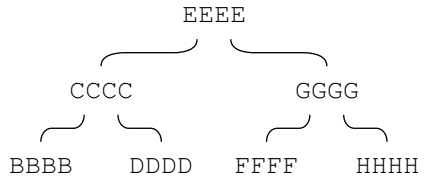
Mamy dane drzewo binarne

```
#define DLUGOSC 32

typedef struct wezel_ {
    char tekst[DLUGOSC];
    struct wezel_ *lewy, *prawy;
} wezel;

wezel *korzen;
```

Napisać funkcję, która wyświetla na ekranie zawartość tego drzewa z zachowaniem struktury. Przykładowo dla drzewa prezentowanego poniżej po lewej, winniśmy otrzymać wynik pokazany po prawej.




```
EEEE
|
+-- CCCC
|   |
|   +-- BBBB
|   |   |
|   |   +-- AAAA
|   |
|   +-- DDDD
|
+-- GGGG
    |
    +-- FFFF
    |
    +-- HHHH
```