3 Tablice i wskaźniki

> Understanding pointers in C is not a skill, it's an aptitude.
>
> Joel Spolsky

W zadaniach z tego zestawu zakładamy, że czytelnik zapoznał się z pojęciem tablicy, zarówno jedno- jak też wielowymiarowych oraz opanował arytmetykę wskaźników. Materiał niezbędny do rozwiązania tych zadań zawiera się w rozdziale piątym [2].

0x16 Zadanie

Napisać program „rysujący" w trybie tekstowym histogram częstości wystąpień poszczególnych liter alfabetu w ciągu wejściowym.

0x17 Zadanie

Napisać program sprawdzający czy dwa słowa wpisane przez użytkownika są anagramami, to znaczy jedno z nich powstaje z drugiego przez pewną permutację liter.

0x18 Zadanie

Napisać program zmieniający liczbę naturalną podaną przez użytkownika na postać słowną.

0x19 Zadanie

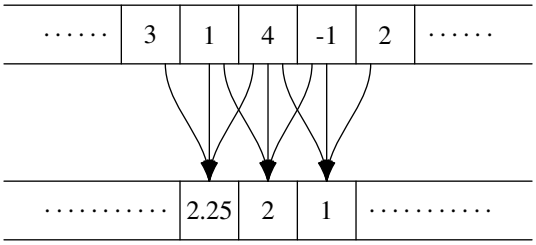
Mamy daną tablicę

```
#define ROZMIAR 100
double sygnal[ROZMIAR];
```

Napisać funkcję przyjmującą jako argument tablicę trójelementową `K` liczb zmiennoprzecinkowych i dokonującą splotu tablicy `sygnal` z tablicą `K`, to znaczy zastępującą każdy element z `sygnal` średnią ważoną jego i jego sąsiadów z wagami zadanymi przez `K`. Przykładowo jeśli fragment tablicy ma postać

·····	3	1	4	-1	2	·····

zaś tablica `K` zawiera elementy 0.25, 0.5, 0.25, to w wyniku winniśmy otrzymać:



0x1A Zadanie

Liczby naturalne 256-bitowe możemy reprezentować za pomocą typu:

```
#define ROZMIAR 32
typedef uint8_t naturalna[ROZMIAR];
```

Zaimplementować funkcje dodawania i mnożenia tych liczb.

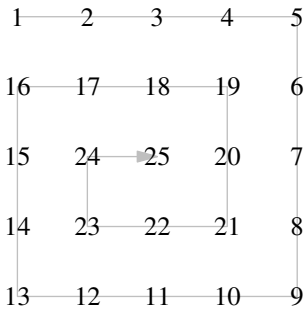
0x1B Zadanie ♣

Znaleźć i poprawić błąd w następującej deklaracji tablicy

```
double macierz[4, 3] = {
    {10,10,10}, {10,0,0}, {0,0.1,0.1}, {0,0,1.1}
};
```

0x1C Zadanie

Napisać program wypełniający kwadratową tablicę kolejnymi liczbami naturalnymi „po spirali", przykładowo dla wymiaru 5 winniśmy uzyskać



0x1D Zadanie

Dana jest tablica

```
#define ROZMIAR 100
bool tab[ROZMIAR][ROZMIAR];
```

Napisać funkcję przyjmującą jako argumenty współrzędne x, y elementu tej tablicy i wypełniającą jedynkami maksymalną składową 4-spójną zawierającą ten element i ograniczoną też jedynkami.

0x1E Zadanie ★

Napisać funkcję obracającą *in situ* macierz kwadratową o $90°$.

In situ

Przekształcenie tablicy (bądź innej struktury agregującej dane) *in situ* oznacza, iż operacje są wykonywane „w miejscu", bezpośrednio na elementach tablicy. Nie jest tworzona kopia tablicy, a rozmiar używanych zmiennych tymczasowych nie zależy od jej rozmiaru.

0x1F Zadanie ☠

Przeanalizować następującą funkcję:

```
void zadanie(int * restrict p, int *restrict q)
{
    *p ^= *q;
    *q ^= *p;
    *p ^= *q;
}
```

0x20 Zadanie

Znaleźć i poprawić błąd w następującym programie

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int dzialanie(int x, int y)
5  {
6      return ((x+y)*abs(x-y)) >> 1;
7  }
8
9  int (*funkcja)(int, int);
10
11 int main(void)
12 {
13     int a = 1005, b = 10;
14
15     funkcja = dzialanie + abs(a);
16     printf("%d, %d", dzialanie(a,b), funkcja(a,b));
17     return 0;
18 }
```

0x21 Zadanie

Znaleźć i poprawić błędy w następującej funkcji, służącej (w założeniu) do odwracania tablicy złożonej z elementów dowolnego typu.

```
1  /*
2  Funkcja odwraca kolejność elementów w tablicy,
3  pTab  - wskaźnik na początek tablicy
4  iRozm - ilość elementów w tablicy
5  iKrok - rozmiar pojedynczego elementu
```

```
6  */
7  void odwrocTablice(void *tab, int rozm, int krok)
8  {
9      void *p, *q;
10
11     p = tab;
12     q = &p + rozm*krok;
13     while( p < q ) {
14         *p++ = *q--;
15     }
16 }
```