

6 Zadania różne

Układamy listę, która ma zawierać wszystkie możliwe imiona Boga.

A. Clarke, *Dziewięć miliardów imion Boga*

Zakładamy, że czytelnik, który dotarł do tego zestawu, poznał już wszystkie elementy języka C i jego biblioteki standardowej. Zadania z tego zestawu mają więc na celu wyćwiczenie i utrwalenie nabytych umiejętności.

0x36 ZADANIE

W opowiadaniu Artura Clarke’a „Dziewięć miliardów imion Boga” z roku 1955 specjalnie przystosowany komputer miał za zadanie wygenerować wszystkie słowa złożone z nie więcej niż dziewięciu liter z wyłączeniem słów zawierających trzy (lub więcej) kolejne takie same litery. Słowa te miały jakoby zawierać wszystkie prawdziwe imiona Boga. W opowiadaniu zatrudniony do tego celu komputer wykonał pracę w ciągu trzech miesięcy. Zaimplementować program realizujący to samo zadanie, zmierzyć jaki czas jest na to potrzebny współczesnym komputerom¹⁾. Sprawdzić czy słów takich jest istotnie dziewięć miliardów.

0x37 ZADANIE ★

Przeanalizować następującą funkcję:

```
unsigned zadanie(unsigned x)
{
    unsigned maskaG, maskaD, przesuniecie;

    przesuniecie = 4*sizeof(unsigned);
    maskaG = ~0;
    maskaD = maskaG >> przesuniecie;
    maskaG = maskaD << przesuniecie;

    do {
```

¹⁾W opowiadaniu wypisanie wszystkich imion Boga okazało się być celem istnienia ludzkości. Zakończenie działania programu spowodowało koniec świata. Zadanie to zatem czytelnik wykonuje na własną odpowiedzialność.

Odwrotna Notacja Polska

Odwrotna Notacja Polska (ang. *Reverse Polish Notation* w skrócie RPN) jest to notacja, w której operator arytmetyczny występuje *po* swoich argumentach. Na przykład wyrażenie (zapisane w notacji tradycyjnej) „2 + 2” przyjmuje w notacji RPN postać „2 2 +”. W przeciwieństwie do notacji tradycyjnej, notacja RPN nie wymaga użycia nawiasów ani określania priorytetów operatorów. Przykładowo wyrażenie

$$2 + 3 \cdot (7 + 4) / (5 - 2)$$

przyjmuje w notacji RPN postać:

$$2\ 3\ 7\ 4\ +\ 5\ 2\ -\ /\ \cdot\ +$$

```

przesuniecie >= 1;
if( x & maskaG ) {
    x &= maskaG;
    maskaD = maskaG & (maskaG >> przesuniecie);
    maskaG ^= maskaD;
} else {
    maskaG = maskaD & (maskaD << przesuniecie);
    maskaD ^= maskaG;
}
} while( przesuniecie > 0 );

return x;
}

```

0x38 ZADANIE

Mamy daną tablicę:

```

#define SZEROKOSC 800
#define WYSOKOSC 600

uint32_t obraz[SZEROKOSC*WYSOKOSC];

```

Elementy tej tablicy interpretujemy jako czwórki liczb typu `uint8_t` określających składowe koloru RGBA. Napisać makropolecenia pozwalające na indeksowanie tablicy `obraz` za pomocą współrzędnych `x`, `y` oraz zwracające wartości poszczególnych składowych koloru.

0x39 ZADANIE

Napisać funkcję obliczającą wartość wyrażenia arytmetycznego zapisanego w notacji RPN.

0x3A ZADANIE

Napisać funkcję konwertującą wyrażenie arytmetyczne z notacji RPN na notację natu-

IOCCC

The International Obfuscated C Code Contest (w skrócie IOCCC), czyli Międzynarodowy Konkurs na Najbardziej Zamroczony Kod C, jest to organizowany od 1984 coroczny (niemal) konkurs na najbardziej nieczytelny program napisany w języku C. Celem konkursu jest między innymi podkreślenie w ironiczny sposób znaczenia standardów kodowania i czytelności programów. Prezentowany w zadaniu 0x3F program, w porównaniu z laureatami IOCCC, można traktować jako „całkiem czytelny”. Oficjalna strona konkursu znajduje się pod adresem <http://www.ioccc.org/>.

ralną.

0x3B ZADANIE

Napisać funkcję konwertującą wyrażenie arytmetyczne z notacji naturalnej na notację RPN.

0x3C ZADANIE

Stworzyć program obliczający wartość wyrażenia zapisanego w notacji tradycyjnej.

0x3D ZADANIE

Napisać kalkulator macierzowy pozwalający na wykonywanie działań na macierzach kwadratowych o wymiarze 2×2 .

0x3E ZADANIE

Napisać program, który porównuje dwa pliki tekstowe w poszukiwaniu pierwszej pary identycznych wierszy.

0x3F ZADANIE

Korzystając ze specyfikacji języka C, przeanalizować poniższy kod²⁾

```

1 #define C(c) ??=c ??=c
2 #define PP C(+)
3 #define IA(a) int a
4 #define CA(a) char a
5
6 IA(puts)();
7 IA(main)(argc, argv)
8 IA(argc);
9 CA(**argv);
10 ??<
11 CA(sT) ??( ??)
12 = C(C) C(a) "nsz  "
13 PP;
14 IA(aP) ??( ??)
15 = ??<
16 6, 3, 5, 4, 8, 1

```

²⁾Przedstawiony kod programu jest *formalnie poprawnym* programem w języku C! Każdy kompilator zgodny ze standardem języka musi go skompilować, jakkolwiek powinien wygenerować ostrzeżenia.

```
17  ??>;
18  IA(i)=0;
19  do ??<
20  sT??(
21  aP??(
22  i??)
23  ??)=
24  sT??(
25  aP??(
26  i+1??)
27  ??);
28  ??>
29  while(++i < 5);
30      sT??(
31  l??)
32  = 'z';
33  puts(sT);
34  return 0;
35  ??>
```