Feladat

Állítsuk elő egy természetes szám decimális (tízes számrendszerbeli) alakjának számjegyeit helyi érték szerint csökkenő sorrendben, egy vektorban!

Specifikáció

$$\begin{array}{lcccccc} A & = & \mathbb{N} & \times & V_{\mathbb{N}} & (\times & \mathbb{Z}) \\ & & n & & v & & k \\ B & = & \mathbb{N} & \times & V_{\mathbb{N}} & & \\ & & n' & & v' & & \end{array}$$

$$\begin{array}{lcl} Q & = & n' = n \wedge v' = v \wedge b' \geq 2 \wedge v.lob = 0 \wedge v.dom >= \lfloor \log_{10} n \rfloor + 1 \\ R & = & v.lob = v'.lob \wedge v.hib = v'.hib \wedge \\ & & \forall i \in [v.lob, v.hib] : v_i \in \{0, \ldots, 9\} \wedge \sum_{i=v.lob}^{v.hib} 10^i v_i = n' \end{array}$$

Absztrakt program

Az alábbi ciklus Q'-n keresztül megoldja a feladatot:

$$\begin{array}{lcl} Q' & = & Q \wedge k = v.hib + 1 \\ P & = & v.lob = v'.lob \wedge v.hib = v'.hib \wedge \\ & & n = n' \div 10^{\lfloor \log_{10} n' \rfloor - k} \wedge \\ & & \sum_{i=k}^{v.hib} v_i = n' \mod 10^{v.hib-k} \wedge \\ & & \forall i \in [k, v.hib] : v_i \in \{0, \ldots, 9\} \\ t & = & k - v.lob \\ \pi & = & n \neq 0 \end{array}$$

Ahol $\div$ jelöli az egészosztást. Bizonyítás nélkül közöljük az alábbi programot, amely a fenti ciklus segítségével megoldja a feladatot:

$k := v.hib + 1$	
$k \neq v.lob$	
	$v[k-1] := n \mod 10$
	$n := n \div 10$
	$k := k - 1$

Fekete doboz-tesztelés

1. Tetszőleges természetes szám számjegyeinek meghatározása
2. A 0 számjegyeinek meghatározása
3. Túl kicsi méretű, esetleg nem nullától címzett vektor megadása az eredmény tárolása számára

(Megjegyzendő, hogy annak a fekete doboz-tesztelés keretében nincs értelme, hogy pl. „negatív szám megadása”, mivel az absztrakt program, ill. a feladat szintjén nincs értelme az állapottéren kívül eső pontokról beszélni; ennyi erővel azt is tesztelhetnénk, hogyan viselkedik a program, ha n értéke a bengáli tigrisek összes részhalmazának halmaza)

C++ implementáció

```
#include <cassert>

// Nem használunk C++ osztályokat, mert még nem tartunk ott az
// anyagban :)
// Emlékezzünk vissza rá, hogy a vektoroknak fix mérete van a
// relációs modellünkben, ezért ilyen a reprezentáció, és ezért
// kell kikötéseket tennünk a méretére a programunkban
struct termeszetes_vektor
{
    int lob;
    int hib;
    unsigned int *val;
};

// Ennyi a lényeg, a többi sallang
void szamjegyek (unsigned int n, termeszetes_vektor &v)
{
    // Érvényes a vektor?
    assert (v.lob == 0 && v.hib >= v.lob && v.val);

    int k;

    k = v.hib + 1;
    while (k != v.lob)
    {
        v.val[k - 1] = n % 10;
        n = n / 10;
        k = k - 1;
    }

    // Elég nagy (volt) a vektor?
    assert (n == 0);
}

// Innentől gyakorlatilag csak a teszt-framework jön
#include <iostream>

unsigned int termeszetes_beolvas ()
{
    int n = -1;

    std::cin >> n;
    if (n < 0)
    {
        std::cerr << "HIBA! Természetes számot adj meg!" << std::endl;
        exit (1);
    }

    return n;
}

int main (int argc, char **argv)
{
    std::cout << "Decimalis szamjegyekre bontas" << std::endl;

    std::cout << "A felbontando szam?" << std::endl;
```

```
    const unsigned int n = termeszetes_beolvas ();

    // A 32 bites, előjel nélküli számnak bőven elég lesz a
    // 12 decimális jegy
    termeszetes_vektor v;
    v.lob = 0;
    v.hib = 11;
    v.val = new unsigned int[v.hib - v.lob + 1];

    szamjegyek(n, v);

    // Eredmény kiírása
    for (int i = v.lob; i <= v.hib; ++i)
        std::cout << v.val[i] << std::endl;

    return 0;
}
```

Fehér doboz-tesztelés

1. Nem természetes szám megadása (pl. negatív szám, sztring)
2. Nem érvényes (pl. nulla méretű) vektor használata
3. A bevitt szám számjegyeinek tárolásához nem elegendő méretű vektor használata