

Feladat

Egy vektor egy természetes szám decimális (tíz-es számrendszerbeli) alakjának számjegyeit tartalmazza helyiérték szerint csökkenő sorrendben. Állítsuk elő ugyanebben a vektorban az eredetnél eggyel nagyobb szám decimális számjegyeit, és mondjuk meg volt-e túlsordulás (amikor az új szám decimális számjegyei nem férnek el a vektorban)!

Specifikáció

$V = \text{vect}(Z, \{0,1, \dots, 9\})$

$A = V \times N_0 \times Z \times L \times N$
 $\text{tmb} \quad h \quad \text{lastValid} \quad \text{ell} \quad n$

$B = V \times N$
 $\text{tmb}' \quad n'$

$Q = (\text{tmb} = \text{tmb}' \wedge \forall i \in [\text{tmb.lob}, \text{tmb.hib}] : (\text{tmb}[i] \in [0,9]) \wedge n = n' \wedge n > 0)$

$R = (Q \wedge \sum_{i \in [\text{tmb.lob}, \text{tmb.hib}]} 10^{\text{tmb.hib}-i} (\text{tmb}[i] + 1) \chi(o))$

Absztrakt program

$h, \text{ell} := \text{tmp.lob} + 1, \text{hamis}$	
$h \neq n$	
ell	
$\text{lastValid} := h$	SKIP
$h := h + 1$	
$\text{tmb}[\text{lastValid} - 1] := \text{tmb}[\text{lastValid} - 1] + 1$	

Tesztelési terv

Tesztesetek a feladat alapján (fekete doboz tesztelés)

- 0-s tmb megadása lehetséges-e.
- A vektor számjegyeinek helyes megadása, csak 0-9 között lehet.

Tesztesetek a kód alapján (fehér doboz tesztelés)

- Olyan szám adatok (helyes és hibás) megadása, amellyel a beolvasó ellenőrző ágát letesztelhetjük.
- Hibás adat esetén, új adat bekérése
- Túlsordulás lekezelés, hiba üzenettel.

Megoldás C++

```
#include <iostream>
#include <string>
#include <conio.h>
using namespace std;

int getNum(bool end); //Ellenörzőm, hogy csak számot írjon be a flsz.
megadhatom, hogy többkaraktert, vagy csak let vihet fel! Ha több számjegy is
lehet, akkor 0-t megengedem, ha csak 1 számjegy, akkor 0 nincs megengedve!
int StackFull(int tmp); //9 + 1 túlcsordul.. ilyenkor -1 a visszatérési értéke
int checkNum(string tmp); //csak szám lehet! itt végzem el a string
vizsgálatát
void putResult(int tmb[], int tmbCount); //eredmény kiíratás, tmp több elemu
tomb, és a mérete -> sizeof al si lehetne számolni a tanultak alapján, de..
így a nyerő.
//-----
int main()
{
    cout << "Adja meg a tomb hosszat (1 <= n )" << endl;
    int n = getNum(true);
    int* tmb = new int[ n ];

    for(int k = 0; k < n; k++)
    {
        cout << "Adja meg a helyi értékeket := ";
        tmb[ k ] = getNum(false);
    }
    putResult(tmb, n);
    int lastValid = 0;
    int ell = 0;
    for(int h = 1; h <= n; h++)
    {
        ell = StackFull(tmb[ h - 1 ] + 1);
        if(ell != -1)
        {
            lastValid = h;
        }
    }
    if(lastValid != 0)
    {
        tmb[ lastValid - 1 ] = tmb[ lastValid - 1 ] + 1;
        for(int null = lastValid; null < n; null++)
        {
            tmb[ null ] = 0;
        }
        putResult(tmb, n);
    }
    else
    {
        cout << "game over!" << endl; //a túlcsordult elem
    }
    getch();
    return 0;
} //int main()
//-----
void putResult(int tmb[], int tmbCount)
{
    cout << "\nTomb erteke = ";
    for(int z = 0; z < tmbCount; z++)
    {
        cout << tmb[ z ]; //a túlcsordult elem
    }
} //void putResult(int tmb[], int tmbCount)
//-----
```

```

//-----
int StackFull(int tmp)
{
    if(tmp > 9 )
    {
        //cout << "Tul nagy a megadott: " << tmp << " erteke!" << endl; //a tulcsordult elem
        return -1;
    }
    else
    {
        return tmp;
    }
} //int StackFull(unsigned int tmp)
//-----
int checkNum(string tmp)
{
    for(int h = 0; h < (signed)tmp.length(); h++)
    {
        if(tmp[ h ] >= '0' && tmp[ h ] <= '9')
        {
            //hmm
        }
        else
        {
            cout << "hiba! tartalmaz mas erteket is!" << endl;
            return -1;
        }
    }
    int n = atoi(tmp.c_str());
    return n;
} //int checkNum(string tmp)
//-----
int getNum(bool end)
{
    string str;
    int n = 0, ell = 1;
    do
    {
        if(end == true)
        {
            cout << "Tomb hossza = ";
        }
        cin >> str;
        if(end == true)
        {
            n = checkNum( str );
            ell = 1;
        }
        else if(end == false)
        {
            if( 1 < str.length())
            {
                cout << "Kerem adja meg a szamot (ugyeljen, hogy 1 <= n <= 9!)" << endl;
                n = -1;
            }
            else
            {
                n = checkNum( str );
                ell = 0;
            }
        }
    } while( !(ell <= n));
    return n;
} //int getNum()

```