EAF I Feladat dokumentáció a 12. házi feladathoz

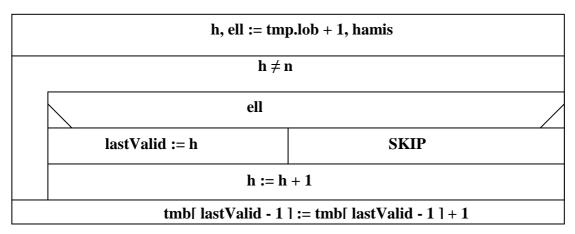
Feladat

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

Specifikáció

$$\begin{split} V &= vect(Z, \{0,1,\cdots,9\}) \\ A &= V \times N_0 \times Z \quad \times \quad L \quad \times \quad N \\ &\quad tmb \quad h \quad lastValid \quad ell \quad \quad n \\ B &= V \quad \times \quad N \\ &\quad tmb' \quad n' \\ Q &= (tmb = tmb' \land \forall \ \forall i \in [tmb.lob,tmb.hib] : (tmb[\ i\] \in [\ 0,9\]) \land n = n' \land n > 0) \\ R &= (Q \land \Sigma_{i \in [tmb.lob,tmb.hib]} \ 10^{tmb.hib-i} (tmb[\ i\] + 1) \ \chi(o)) \end{split}$$

Absztrakt program



Tesztelési terv

Tesztesetek a feladat alapján (fekete doboz tesztelés)

- 1. 0-s tmb megadása lehetséges-e.
- 2. A vektor számjegyeinek helyes megadása, csak 0-9 között lehet.

Tesztesetek a kód alapján (fehér doboz tesztelés)

- 1. Olyan számadatok (helyes és hibás) megadása, amellyel a beolvasó ellenőrző ágát letesztelhetjük.
- 2. Hibás adat esetén, új adat bekérése
- 3. Túlcsordulás lekezelés, hiba üzenettel.

Megoldás C++

```
#include <iostream>
#include <string>
#include <conio.h>
using namespace std;
int getNum(bool end); //Ellenörzöm, hogy csak számot írjon be a flsz.
megadhatom, hogy tobbkaraktert, 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;</pre>
      int n = getNum(true);
      int* tmb = new int[ n ];
      for(int k = 0; k < n; k++)
            cout << "Adja meg a helyiertekeket := ";</pre>
            tmb[ k ] = getNum(false);
      putResult(tmb, n);
      int lastValid = 0;
      int ell = 0;
      for(int h = 1; h <= n; h++)</pre>
            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++)</pre>
                  tmb[null] = 0;
            putResult(tmb, n);
      }
      else
            cout << "game over!" << endl; //a tulcsordult elem</pre>
getch();
return 0;
}//int main()
//----
void putResult(int tmb[], int tmbCount)
      cout << "\nTomb erteke = ";</pre>
      for(int z = 0; z < tmbCount; z++)
            cout << tmb[ z ]; //a tulcsordult elem</pre>
}//void putResult(int tmb[], int tmbCount)
```

```
int StackFull(int tmp)
       if(tmp > 9)
              //cout << "Tul nagy a megadott: " << tmp << " ertek!" << 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++)</pre>
              if(tmp[ h ] >= '0' && tmp[ h ] <= '9')</pre>
                      //hmm
              }
              else
                      cout << "hiba! tartalmaz mas erteket is!" << endl;</pre>
                     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;
              if(end == true)
                     cout << "Tomb hossza = ";</pre>
              cin >> str;
              if(end == true)
                     n = checkNum( str );
                     ell = 1;
              else if(end == false)
                      if( 1 < str.length())</pre>
                             \verb|cout| << | "Kerem adja meg a szamot (ugyeljen, hogy 1 <= n és n <= 9!)" << endl; \\
                      }
                      else
                             n = checkNum( str );
                             ell = 0;
       }while( !(ell <= n));</pre>
return n;
}//int getNum()
```