

1. feladat:

Az alábbi algoritmus egy szövegnek meghatározza a leghosszabb szavát és annak hosszát. Írja át időben hatékonyabbra! Írja meg az időben hatékonyabb Pascal programot!

Eljárás Maxkiv(S:Szöveg; **Változó** Lhszo:Szöveg; Mhossz:Egész):

```
Lhszo:=''; Szo:=''
Ciklus i=1-től hossz(S)-ig
  Ha S[i]≠' '
    akkor Szo:=Szo+S[i]
    különb Ha hossz(Szo)>hossz(Lhszo) akkor Lhszo:=Szo
      Szo:=''
  Elágazás vége
Ciklus vége
Ha hossz(Szo)>hossz(Lhszo) akkor Lhszo:=Szo
Mhossz:=hossz(Lhszo)
```

Eljárás vége.

2. feladat:

Adott egy egészekből álló alsó háromszög mátrix (vagyis a fő diagonális felett csupa 0 áll). Keressen meg egy olyan oszlopot, amelyre igaz, hogy elemei összegének abszolút értékét mod 2 véve 0-t kapunk, a mögötte levő oszlop elemei összegének abszolút értékét mod 2 véve pedig 1-t kapunk, vagy fordítva (az első ad 1-t, a mögötte lévő pedig 0-t)!

Eljárás Lassú_Keresés(N:Egész, X:TömbTip, **Változó** Van:Logikai, Hol:Egész):

```
Van:=Hamis
Ciklus j=1-től N-1-ig
  Ha (((Oszloposzeg(j) mod 2)=0) és ((Oszloposzeg(j+1) mod 2)=1))
    vagy
    (((Oszloposzeg(j) mod 2)=1) és ((Oszloposzeg(j+1) mod 2)=0))
    és nem Van
    akkor Van:=Igaz; Hol:=j
  Elágazás vége
Ciklus vége
```

Eljárás vége.

Függvény Oszloposzeg(j):egész

```
s:=0
Ciklus i=1-től N-ig
  s:=s+X[i,j]
Ciklus vége
Ha s>=0 akkor Oszloposzeg:=s különb Oszloposzeg:=(-1)*s
```

Függvény vége

Írja át a fenti eljárást időben hatékonyabbra! Írja meg az időben hatékonyabb Pascal programot!

Értékelés: Mindkét feladatra igaz, hogy több helyen lehet hatékonyabbra írás bennük, ezek megvalósítási számától (és helyességétől) függ az osztályzat. Ha csak hatékonyabb algoritmust ír, és papíron beadja, akkor amennyiben az helyes, feladatonként 1-1 osztályzatnyi lehet maximum az értéke, ha helyes programot is (csak a hatékony algoritmusra), akkor feladatonként újabb 1-1 osztályzat lehet maximum. Elégtelenből indulva +4 osztályzattal jöhet ki az 5-ös.

Munkája végétével a [\\inf.elte.hu\dfs\zh\kozinf1](https://inf.elte.hu/dfs/zh/kozinf1) oldalra másolja fel a saját Neptun kódjára keresztelt tömörített állományt, melyben legyenek benne a hatékony programjai (feladat1.pas, feladat2.pas).

Megoldások

1. feladat:

Egy olyan megoldást nézünk, amelyben „mérjük” is a hatékonyság alakulását az eredeti és a javított esetében. Amit érdekes lehet számlálni az az karakterhasonlítások száma, szövegbővítések száma és a szövegmásolások száma.

Program Zh12_13_2_2_1;

```
{  
  Szerző: Szlávi Péter  
  Neptun-kód: ZDOF7J
```

Feladat (2012/13 2. félév 2. zh. 1. feladat):

Az alábbi algoritmus egy szövegnek meghatározza a leghosszabb szavát és annak hosszát. Írja át időben hatékonyabbra!
Írja meg az időben hatékonyabb Pascal programot!

Eljárás Maxkiv(S:Szöveg; Változó Lhszo:Szöveg; Mhossz:Egész):

```
Lhszo:=''; Szo:=''  
Ciklus i=1-től hossz(S)-ig  
  Ha S[i]<>' '  
    akkor Szo:=Szo+S[i]  
    különben Ha hossz(Szo)>hossz(Lhszo) akkor Lhszo:=Szo  
    Szo:=''  
  Elágazás vége  
Ciklus vége  
Ha hossz(Szo)>hossz(Lhszo) akkor Lhszo:=Szo  
Mhossz:=hossz(Lhszo)  
Eljárás vége.
```

Megoldás algoritmus:

Eljárás Maxkiv(S:Szöveg, Változó Lhszo:Szöveg, Mhossz:Egész);

```
Lhszo:=''  
hS:=Length(S)//amígos ciklusnál esetében csak egyszer kelljen kiszámolni  
i:=1  
Ciklus amíg i<=hS  
  //a szóközintervallumok átlépése:  
  Ciklus amíg i<=hS és S[i]=' '  
    i:=1  
  Ciklus vége  
  Ha i<=hS akkor//szókezdetnél tartunk, jöhet a szó:  
    Szo:=S[i]  
    i:=1  
    Ciklus amíg i<=hS és S[i]<>' '  
      Szo:=Szo+S[i]  
      i:=1  
    Ciklus vége  
  Ha i<=hS akkor//szóköznel tartunk, lépjük át:  
    i:=1  
  Elágazás vége  
  //az eddigi leghosszabb-e:  
  Ha hossz(Szo)>hossz(Lhszo) akkor Lhszo:=Szo  
  Elágazás vége  
Ciklus vége  
Mhossz:=Length(Lhszo)  
Eljárás vége.  
}
```

Uses

Crt;

Var

S:String;

Lhszo:String;

Mhossz:Integer;

```
Procedure Beolvas (Var S:String);
Begin
  Writeln('A mondat:'); Readln(S);
End;

//a hatékonyság mérés kedvéért számláljuk a jellegzetes műveleteket:

Var
  azDb,hozzaDb,mozgDb:Integer;

Procedure Inic;
Begin
  azDb:=0; hozzaDb:=0; mozgDb:=0;
End;

Function AzonosE(Const c1,c2:Char):Boolean;
Begin
  AzonosE:=c1=c2;
  Inc(azDb);
End;

Procedure KarakterMozgatas (Var s1:String; Const s2:String);
Begin
  s1:=s2;
  Inc(mozgDb,Length(s2)+1);//+1: hossz-állítás
End;

Procedure Hozza (Var s:String; Const c:Char);
Begin
  s:=s+c;
  Inc(hozzaDb,Length(s)+1);//+1: hossz-állítás
End;

//Eredeti kód: -----

Procedure Maxkiv(S:String; Var Lhszo:String; Var Mhossz:Integer);
  Var
    i:Integer;
    Szo:String;
Begin
//  Lhszo:=''; Szo:='';
  KarakterMozgatas(Lhszo,''); KarakterMozgatas(Szo,'');
  For i:=1 to Length(S) do
  Begin
//    If S[i]<>' ' then
      If Not AzonosE(S[i],' ') then
      Begin
//        Szo:=Szo+S[i];
        Hozza(Szo,S[i]);
      End
      else
      Begin
//        If Length(Szo)>Length(Lhszo) then Lhszo:=Szo;
        If Length(Szo)>Length(Lhszo) then KarakterMozgatas(Lhszo,Szo);
//        Szo:='';
        KarakterMozgatas(Szo,'');
      End
    End;
  End;
//  If Length(Szo)>Length(Lhszo) then Lhszo:=Szo;
  If Length(Szo)>Length(Lhszo) then KarakterMozgatas(Lhszo,Szo);
  Mhossz:=Length(Lhszo);
End;

//Hatékonyabb kód: -----

Procedure MaxkivGyors(S:String; Var Lhszo:String; Var Mhossz:Integer);
  Var
    i,hS:Integer;
    Szo:String;
```

```

Begin
//   Lhszo:='';
   KarakterMozgatas(Lhszo,'');
   hS:=Length(S); //amígos ciklusnál előre, csak egyszer kell kiszámolni
   i:=1;
   While i<=hS do
   Begin
     //a szóközintervallumok átlépése:
     //   While (i<=hS) and (S[i]=' ') do
     While (i<=hS) and AzonosE(S[i],' ') do
     Begin
       Inc(i);
     End;
     If i<=hS then //szókezdetnél tartunk, jöhet a szó:
     Begin
//       Szo:=S[i];
       KarakterMozgatas(Szo,S[i]);
       Inc(i);
//       While (i<=hS) and (S[i]<>' ') do
       While (i<=hS) and Not AzonosE(S[i],' ') do
       Begin
         Szo:=Szo+S[i];
         Hozza(Szo,S[i]);
         Inc(i);
       End;
       If i<=hS then //szóköznel tartunk, lépjük át:
       Begin
         Inc(i)
       End;
//az eddigi leghosszabb-e:
//       If Length(Szo)>Length(Lhszo) then Lhszo:=Szo;
       If Length(Szo)>Length(Lhszo) then KarakterMozgatas(Lhszo,Szo);
     End;
   End;
   Mhossz:=Length(Lhszo);
End;

Begin
  ClrScr;
  Beolvas(S);
  Inic;
  Maxkiv(S, Lhszo,Mhossz);
  Writeln('A szöveg "',S,'" hossza:',Length(S),'.');
  Writeln('A leghosszabb szó "',Lhszo,'" hossza:',Mhossz,'.');
  Writeln(' Azonosságvizsgálatok száma:',azDb);
  Writeln(' Hozzávételek száma:',hozzaDb);
  Writeln(' Karaktermozgatások száma:',mozgDb);
  ReadKey;
  Inic;
  MaxkivGyors(S, Lhszo,Mhossz);
  Writeln('A szöveg "',S,'" hossza:',Length(S),'.');
  Writeln('A leghosszabb szó "',Lhszo,'" hossza:',Mhossz,'.');
  Writeln(' Azonosságvizsgálatok száma:',azDb);
  Writeln(' Hozzávételek száma:',hozzaDb);
  Writeln(' Karaktermozgatások száma:',mozgDb);
  ReadKey;
End.

```

2. feladat:

A megoldás algoritmus:

Eljárás Gyors_Keresés(N:Egész, X:TömbTip, **Változó** Van:Logikai, Hol:Egész):

```

j:=1
oÖA0:=Oszloposzeg(1) mod 2=0 //az aktuális oszlop összege páros-e?
oÖK0:=Oszloposzeg(2) mod 2=0 //a következő oszlop összege páros-e?
Ciklus amíg j<N és (nem oÖA0 vagy oÖK0) és (oÖA0 vagy nem oÖK0)
  j:=j+1
  oÖA0:=oÖK0
  oÖK0:=Oszloposzeg(j+1) mod 2=0

```

Ciklus vége

Van:=k<N

Ha Van akkor Hol:=j**Eljárás vége.****Függvény** Oszloposzeg(j) :**egész**

s:=0

Ciklus i=j-tól N-ig

s:=s+X[i,j]

Ciklus vége~~**Ha** s>=0 **akkor** Oszloposzeg:=s **különben** Oszloposzeg:=(-1)*s~~**Függvény vége**

A kód egy olyan megoldáskódja, amelyben „mérjük” a hatékonyság alakulását is az eredeti és a javított esetében. Amit érdekes lehet számlálni az most az összeadások száma.

Program Zh12_13_2_2_2;

{

Szerző: Szlávi Péter

Neptun-kód: ZDOF7J

Feladat (2012/13 2. félév 2. zh. 2. feladat):

Adott egy egészekből álló alsó háromszög mátrix (vagyis a fő diagonális felett csupa 0 áll). Keressen meg egy olyan oszlopot, amelyre igaz, hogy elemei összegének abszolút értékét mod 2 véve 0-t kapunk, a mögötte levő oszlop elemei összegének abszolút értékét mod 2 véve pedig 1-t kapunk, vagy fordítva (az első ad 1-t, a mögötte lévő pedig 0-t)!

Eljárás Lassú_Keresés(N:Egész, X:Tömbtip, Változó Van:Logikai, Hol:Egész):

Van:=Hamis

Ciklus j=1-től N-1-ig**Ha** (((Oszloposzeg(j) mod 2)=0) és ((Oszloposzeg(j+1) mod 2)=1)) vagy

(((Oszloposzeg(j) mod 2)=1) és ((Oszloposzeg(j+1) mod 2)=0))) és nem

Van

akkor Van:=Igaz; Hol:=j

Elágazás vége

Ciklus vége

Eljárás vége.

Függvény Oszloposzeg(j) :**egész**

s:=0

Ciklus i=1-től N-ig

s:=s+X[i,j]

Ciklus vége**Ha** s>=0 **akkor** Oszloposzeg:=s **különben** Oszloposzeg:=(-1)*s**Függvény vége**

Írja át a fenti eljárást időben hatékonyabbra! Írja meg az időben hatékonyabb Pascal programot!

Megoldás algoritmus:

Eljárás Gyors_Keresés(N:Egész, X:Tömbtip, Változó Van:Logikai, Hol:Egész):

j:=1

oÖA0:=Oszloposzeg(1) mod 2=0 //az aktuális oszlop összege páros-e?

oÖK0:=Oszloposzeg(2) mod 2=0 //a következő oszlop összege páros-e?

Ciklus amíg j<N és (nem oÖA0 vagy oÖK0) és (oÖA0 vagy nem oÖK0)

j:=j+1

oÖA0:=oÖK0

oÖK0:=Oszloposzeg(j+1) mod 2=0

Ciklus vége

Van:=j<N

Ha Van **akkor** Hol:=j

Eljárás vége.

```
Függvény Oszloposzeg(j):egész
  s:=0
  Ciklus i=j-tól N-ig
    s:+X[i,j]
  Ciklus vége
  Oszloposzeg:=s
Függvény vége
}

Uses
  Crt;
Const
  MaxN=20;
Type
  TombTip=Array [1..MaxN,1..MaxN] of Integer;
Var
  AHM:TombTip;
  N:Integer;
  VanE:Boolean;
  Hol:Integer;

Procedure Feltolt;
  var
    i,j:Integer;
    c:Char;
Begin
  Write('A méret:'); Readln(N);
  Writeln('Kalviatura-bemenet/Automatikusan? (K/A)');
  c:=ReadKey;
  If UpCase(c)='K' then
  Begin
    For i:=1 to N do
    Begin
      For j:=1 to i do
      Begin
        Write('(',i:3,',',j:3,')'); Readln(AHM[i,j]);
      End;
      For j:=j+1 to N do
      Begin
        AHM[i,j]:=0;
      End;
    End;
  End;
else
Begin
  For i:=1 to N do
  Begin
    For j:=1 to i do
    Begin
      AHM[i,j]:=Random(2*N)-N;
    End;
    For j:=j+1 to N do
    Begin
      AHM[i,j]:=0;
    End;
  End;
End;
  //ellenőrző listázás:
  For i:=1 to N do
  Begin
    For j:=1 to N do
    Begin
      Write(AHM[i,j]:3);
    End;
    Writeln;
  End;
End;
```

```
//a hatékonyság mérés kedvéért számláljuk a jellegzetes műveleteket:
```

```
Var
```

```
  hozzáDb:Integer;
```

```
Procedure Inc;
```

```
Begin
```

```
  hozzáDb:=0;
```

```
End;
```

```
Procedure Hozzaad(Var s:Integer; Const e:Integer);
```

```
Begin
```

```
  s:=s+e;
```

```
  Inc(hozzáDb);
```

```
End;
```

```
//Eredeti kód: -----
```

```
Procedure Lassu_Kereses(N:Integer; X:TombTip; Var Van:Boolean; Var Hol:Integer);
```

```
  Var
```

```
    j:Integer;
```

```
  Function Oszloposzeg(Const j:Integer):Integer;
```

```
    Var
```

```
      s,i:Integer;
```

```
  Begin
```

```
    s:=0;
```

```
    For i:=1 to N do
```

```
      Begin
```

```
//        s:=s+X[i,j];
```

```
        Hozzaad(s,X[i,j]);
```

```
      End;
```

```
      If s>=0 then Oszloposzeg:=s else Oszloposzeg:=(-1)*s;
```

```
    End;
```

```
Begin
```

```
  Van:=False;
```

```
  For j:=1 to N-1 do
```

```
    Begin
```

```
      If
```

```
        (((Oszloposzeg(j) mod 2)=0) and ((Oszloposzeg(j+1) mod 2)=1)) or
```

```
        (((Oszloposzeg(j) mod 2)=1) and ((Oszloposzeg(j+1) mod 2)=0))) and
```

```
        not Van then
```

```
          Begin
```

```
            Van:=True; Hol:=j;
```

```
          End;
```

```
    End;
```

```
End;
```

```
//Hatékonyabb kód: -----
```

```
Procedure Gyors_Kereses(N:Integer; X:TombTip; Var Van:Boolean; Var Hol:Integer);
```

```
  Var
```

```
    j:Integer;
```

```
    oOA0,oOK0:Boolean;
```

```
  Function Oszloposzeg(Const j:Integer):Integer;
```

```
    Var
```

```
      s,i:Integer;
```

```
  Begin
```

```
    s:=0;
```

```
    For i:=j to N do
```

```
      Begin
```

```
//        s:=s+X[i,j];
```

```
        Hozzaad(s,X[i,j]);
```

```
      End;
```

```
      Oszloposzeg:=s;
```

```
    End;
```

```
Begin
```

```
j:=1;
oOA0:=Oszloposzeg(1) mod 2=0;//az aktuális oszlop összege páros-e?
oOK0:=Oszloposzeg(2) mod 2=0;//a következő oszlop összege páros-e?
While (j<N) and (not oOA0 or oOK0) and (oOA0 or not oOK0) do
  Begin
    Inc(j);
    oOA0:=oOK0;
    oOK0:=Oszloposzeg(j+1) mod 2=0;
  End;
Van:=j<N;
If Van then Hol:=j;
End;

Begin
  ClrScr;
  Feltolt;

  Inic;
  Lassu_Kereses(N,AHM, VanE, Hol);
  If VanE then
  Begin
    Writeln('Van, mégpedig a(z) ',Hol, '. oszlopnál. ');
  End
  else
  Begin
    Writeln('Nincs. ');
  End;
  Writeln('Összeadások száma:',hozzaDb);

  Inic;
  Gyors_Kereses(N,AHM, VanE, Hol);
  If VanE then
  Begin
    Writeln('Van, mégpedig a(z) ',Hol, '. oszlopnál. ');
  End
  else
  Begin
    Writeln('Nincs. ');
  End;
  Writeln('Összeadások száma:',hozzaDb);
  ReadKey;
End.
```