

7 knots is 8 MPH... just add one seventh more.

7 knots is 8 MPH... just add one seventh more.

Source: <http://sci.tech-archive.net/Archive/sci.physics/2006-12/msg00363.html>

- *From:* Jeff&Relf <Jeff_Relf@xxxxxxxxx>
 - *Date:* 3 Dec 2006 01:18:24 GMT
-

Hi The_Ghost_In_The_Machine, 7 knots is 8 MPH... just add a seventh more...
Or 10 percent more if yer feelin' lazy... Or Google, " 7 knots in mph "

By the way, it takes a real genius like you to write:

```
bool f(bool g, bool h) { if(g) h = true; else h = false; return h; }
```

It takes a fool like me to write:

Consider diff'ing the strings: " FooBar " vs " BarFoo ".
There are no " normal " differences because
no characters come before the required 3 matching characters (ReSync = 3).
I call this the Swapped_Text condition because
a pair of matching strings merely have swapped positions.
Reporting this Swapped_Text is complicated because:

1. It must be resolved early in the string to prevent compounding errors.
2. End_of_Matching_Sting conditions.
3. End_of_Line conditions.

```
www.Cotse.NET/users/jeffrelf/Dif.PNG  
www.Cotse.NET/users/jeffrelf/Dif.EXE  
( for help, double-click Dif.EXE )  
www.Cotse.NET/users/jeffrelf/Dif.CPP
```

```
#define Set_Prn ( \  
Prn_KlId = Kill.Prn_E - Kill.Prn_B, Prn_Born = Born.Prn_E - Born.Prn_B )
```

```
#define Prn_ReSync( Use ) ( Use.Cmp_B = Use.Cmp_E \  
, Use.Prn_E += ( rv = Use.Forth ? 1 : ReSync ), Use.Cmp_E += rv )
```

```
#define Set_Prn_Cmp_E( Prn, Cmp ) ( \  
Prn_E = er( Prn, EE_Prn ), Cmp_E = er( Cmp, EE_Cmp ) \  
, EoL = Prn_E >= EE_Prn, EoR = Cmp_E >= EE_Cmp \  
, ReSync_EoL = Prn_E + ReSync - 1 >= EE_Prn \  
, ReSync_EoR = Cmp_E + ReSync - 1 >= EE_Cmp )
```

```
#define Inc_Prn_Cmp_E( Prn, Cmp ) \  
Set_Prn_Cmp_E( Prn_E + Prn, Cmp_E + Cmp )
```

7 knots is 8 MPH... just add one seventh more.

7 knots is 8 MPH... just add one seventh more.

```
inline int Is_Apple( wchar_t Ch0, wchar_t Ch, wchar_t Ch2 ) {
return
Ch <= 32 || isdigit( Ch )

inline int Apple_Apple_Eq(
wchar_t Ch0_L, wchar_t Ch_L, wchar_t Ch2_L
, wchar_t Ch0_R, wchar_t Ch_R, wchar_t Ch2_R ) {
if ( ! Ignore_Nums ) return Ch_L == Ch_R ;
int
Lf = Is_Apple( Ch0_L, Ch_L, Ch2_L )
, Rt = Is_Apple( Ch0_R, Ch_R, Ch2_R );
return Lf && Rt || ! Lf && ! Rt && Ch_L == Ch_R ; }

int Eq_Diff( LnP D, LnP L, LnP R ) {
wchar_t Ch0_L, Ch_L, Ch2_L, Ch0_R, Ch_R, Ch2_R ;
if ( D ) { Loop_Until_LR( 1 );
return Apple_Apple_Eq( Ch0_L, Ch_L, Ch2_L, Ch0_R, Ch_R, Ch2_R ); }
Loop_Until_LR( ! Apple_Apple_Eq( Ch0_L, Ch_L, Ch2_L, Ch0_R, Ch_R, Ch2_R ) );
return ! Ch_L && ! Ch_R ; }

int Eq_Diff( LnP D, LnA L, LnA R ) { return Eq_Diff( D, * L, * R ); }

template < typename LnM >
Diff ( LnM BB_Klld, LnM EE_Klld, LnM BB_Born, LnM EE_Born
, wchar_t Ch_Color ) {
struct Prn { int Forth ; LnM Prn_B, Prn_E, Cmp_B, Cmp_E ; }; Prn Kill ;
LnP D = ! Ch_Color ? 0 : _T - 1 ;
if ( D )
(LnP &) EE_Klld = From_Start( LnP( BB_Klld ), 0 ), * _T = 0
, (LnP &) EE_Born = From_Start( LnP( BB_Born ), 0 );
wchar_t Ch_Swapd = Ch_Color == Ch_Klld ? Ch_Klld_Swapd : Ch_Born_Swapd ;
int ReSync = 3, ReSync_EoL, ReSync_EoR, EoL, EoR, Klld = 0 ;
LnM B_Klld = BB_Klld - 1, E_Klld, B_Born = BB_Born - 1, E_Born ;

Set_Prn_Cmp: Klld ^= 1 ;
LnM
BB_Prn = Klld ? BB_Klld : BB_Born, EE_Prn = Klld ? EE_Klld : EE_Born
, BB_Cmp = Klld ? BB_Born : BB_Klld, EE_Cmp = Klld ? EE_Born : EE_Klld
, Prn_B = Klld ? B_Klld : B_Born, Prn_E = Klld ? E_Klld : E_Born
, Cmp_B = Klld ? B_Born : B_Klld, Cmp_E = Klld ? E_Born : E_Klld ;
Set_Prn_Cmp_E( Prn_B, Cmp_B ); int Cnt_Lines = -1, Forth = -1 ;

Was_Eq:
if ( ++ Cnt_Lines == -1 ) {
Forth = ! EoL && ! EoR && Eq_Diff( D, 1 + Prn_E, 1 + Cmp_E );
Inc_Prn_Cmp_E( -ReSync, -ReSync ); goto Set_Recs ; }
if ( Cnt_Lines < 0 ) goto NextPair ;
Prn_B = Prn_E, Cmp_B = Cmp_E ;
if ( Klld && D ) {
if ( Cnt_Lines == 1 ) * ++ D = Ch_Default ;
```

7 knots is 8 MPH... just add one seventh more.

7 knots is 8 MPH... just add one seventh more.

```
if ( Cnt_Lines ) * ++ D = * LnP( Prn_B ); }
if ( EoL && EoR ) {
if ( D || B_Klld >= BB_Klld || B_Born >= BB_Born ) {
if ( D ) * ++ D = 0 ; return ; }
int Ks = Cnt_Lines >= 1000 ;
Str( _T, ! Ks ? L"%d" : L"%d,%03d"
, ! Ks ? Cnt_Lines : Cnt_Lines / 1000, Cnt_Lines % 1000 );
Sh( L"%cIdentical%c, %s Lines.", Ch_Hi, Ch_Default, _T ); return ; }
if ( EoL || EoR ) { Set_Prn_Cmp_E( EE_Prn, EE_Cmp ); goto Set_Recs ; }
NextPair: Inc_Prn_Cmp_E( 1, 1 );
```

```
Cmp: if ( Eq_Diff( D, Prn_E, Cmp_E ) ) goto Was_Eq ;
```

```
Wasnt_Eq: int Back = Cnt_Lines >= 0 ? 0 : 1 + ReSync + Cnt_Lines ;
Inc_Prn_Cmp_E( -Back, -Back );
if ( ! ReSync_EoR ) Inc_Prn_Cmp_E( 0, 1 ); else {
if ( ReSync_EoL ) goto Set_Recs ;
Set_Prn_Cmp_E( Prn_E + 1, Cmp_B + 1 ); }
Cnt_Lines = -1 - ReSync ; goto Cmp ;
```

```
Set_Recs: Prn Born = { Forth, Prn_B, Prn_E, Cmp_B, Cmp_E };
if ( Klld ) { Kill = Born ; goto Set_Prn_Cmp ; }
int Swapd, Prn_Klld , Prn_Born ; Set_Prn;
if ( Swapd = ! Prn_Klld && ! Prn_Born ) {
int Kl = Born.Cmp_E - Kill.Prn_B, Brn = Kill.Cmp_E - Born.Prn_B ;
if ( Kl <= Brn ) Prn_ReSync( Kill );
if ( Kl >= Brn ) Prn_ReSync( Born ); Set_Prn; }
Loop( 1 + ! D ) { int Kl = ! J ;
wchar_t Color = Kl ? Ch_Klld : Ch_Born ;
Loop( Kl ? Prn_Klld : Prn_Born ) {
if ( ! J ) { Prn & R = Kl ? Kill : Born ;
Prn_B = R.Prn_B, Prn_E = R.Prn_E, Cmp_B = R.Cmp_B, Cmp_E = R.Cmp_E ;
if ( D ) * ++ D = Swapd ? Ch_Swapd : Ch_Color ; }
if ( D ) * ++ D = * ++ ( LnP & ) Prn_B ;
else { int Call_Eq = ++ Cmp_B <= Cmp_E ;
if ( * * LnA( ++ Prn_B ) == 1 ) continue ;
if ( Prn_B < EE_Prn, Call_Eq ) {
Diff<LnP>( * LnA( Prn_B ), 0, * LnA( Cmp_B ), 0, Color );
if ( Swapd ) * * LnA( Cmp_B ) = 1 ; }
Sh(
L"%c|%c%s", Color, Call_Eq ? Ch_Default : Color
, Call_Eq ? _T : * LnA( Prn_B ) ); } } }
B_Klld = Kill.Prn_E, B_Born = Born.Prn_E ; goto Set_Prn_Cmp ; }
```

```
Diff_Files( int Wild_Carding, LnP Nam_L, LnP Nam_R ) {
static int F_Num ; F_Num ++ ;
// if ( F_Num != 13 ) return ;
// Ignore_Nums = 1 ;
LnP P = Nam_R ;
if ( Wild_Carding ) {
P = From_Start( P, 0 ) - 1 ;
```

7 knots is 8 MPH... just add one seventh more.

7 knots is 8 MPH... just add one seventh more.

```
if ( * P != '\\ ' && * P != ' ' ) Str( ++ P, L"\\ " ); P = Nam_R ;
Str( P = _T, L"%s%s", Nam_R, Nam_L ); }
Sh( L"%c%c%s%c, %s", Ch_Hi, Nam_L, Ch_Default, Nam_R );
static LnT Lt, Rt ;
Loop( 2 ) { int Klld = ! J ;
LnP Nam = Klld ? Nam_L : P ; LnT & Ln = Klld ? Lt : Rt ;
FILE * fp = fopen ( Nam,L"rb" );
if( ! fp ) { Sh(L"\n_ %s _ Not Found.", Nam ); return ; }
int Sz = filelength( fileno( fp ) );
if( ! Sz ) { Sh(L"\n_ %s _ is empty.", Nam ); return ; }
static __LnP BB[ 2 ];
__LnP & _B = BB[ Klld ]; free( _B );
_B = ( __LnP ) malloc( Sz += 4 * Sz_Char );
__LnP _E = _B + fread( _B, 1, Sz, fp ); fclose( fp );
LnP B = ( LnP ) _B, E = ( LnP ) _E ;
if ( * B != Magic_UTF_16 ) { * _E = 0 ;
B = ( LnP ) malloc( ( Sz + 4 ) * Sz_Char );
Sz = MultiByteToWideChar( CP_ACP, 0, _B, -1, B + 1, Sz );
free( _B ), _B = ( __LnP ) B, E = 1 + B + Sz - 1 ; }
* B ++ = 0 ;
if ( B == E ) { Sh(L"\n_ %s _ is empty.", Nam ); return ; }
if ( * -- E != 10 ) * ++ E = 10 ; * ++ E = 0 ; LnP P = B ;
LnP P_Nxt = From_Start ( P, 0 );
if ( P_Nxt > B && P_Nxt [ -1 ] != 10 ) * P_Nxt = 10, * ++ P_Nxt = 0 ;
Ln.PP = Ln.BB - 1 ;
LOOP { Inc ( Ln ) = P ; LnP P_Nxt = E = From_Start ( P, 10 );
* E = 0 ; while ( * -- E && * E <= 32 ); * ++ E = 0 ;
P = ++ P_Nxt ; if ( ! * P ) break ; } }
Diff<LnA>( Lt.BB, Lt.PP, Rt.BB, Rt.PP, 0 ); }
```

7 knots is 8 MPH... just add one seventh more.