➲ AUFGABE 1:RIBO-NATTER

Lösungsidee und Programmdokumentation:

Die Farbfolge der Ribo-Natter ist im String `RNA` gespeichert. Zu Beginn des Programms wird er ganz einfach eingelesen. Es müssen immer zwei Bereiche miteinander verglichen werden. Darum habe ich zwei ineinander gekoppelte Schleifen, die eine für den `start` des ersten Bereiches und die andere für das `ende` des zweiten Bereiches. In dieser Schleife wird dann die Zählervariable `len` von 0 aus immer erhöht bis einmal die zwei Farben, die vom `Start` bzw. `Ende` um `len` entfernt sind, nicht übereinstimmen. `Len` darf dabei auch nicht größer als die Hälfte des Abstandes zwischen `Start` und `Ende` minus 3 werden. Der Vergleich selbst funktioniert so:
Ich darf davon ausgehen, dass im String nur A, U, C, G, a, u, c und g vorhanden sind. Also ermittle ich immer den Abstand zweier Buchstaben. Für A und U ist dieser 20 und für C und G 4. Da A und G nicht symmetrisch um C oder U liegen, ist der Vergleich eindeutig.
Nach dem das erste mal keine Übereinstimmung zutrifft, wird `len` mit dem Maximum dieses Wertes `maxlen` verglichen. Wenn `len` größer ist, werden für `maxlen`, `maxstart` und `maxende` die momentanen Werte gesetzt.
Um nicht unnütze Start und Endwerte durchzuspielen, bricht die Start-Schleife bereits bei 2*maxlen vor dem wirklichen Ende der Schlange ab und die Ende-Schleife geht auch nicht weiter bis 2*maxlen vor dem Start Wert.

```
PROGRAM RIBO_NATTER;
USES CRT;
CONST stimmig : set of byte = [4, 20];
VAR RNA : string;                                          //Farbfolge der Ribo-Natter
    start, ende, len : word;                          //monentan betrachtete Harmoniestelle
    maxstart, maxende, maxlen : word;                    //vorläufig längste Harmoniestelle

BEGIN
clrscr;
maxlen := 0;
WRITELN('DIE FARBENFOLGE DER RIBONATTER:');
readln(rna);                                                                    //Eingabe

FOR start := 1 to length(rna)-2*maxlen do                    //Schleife für die Startpunkte
 FOR ende  := length(rna) downto start+2*maxlen do BEGIN      //Schleife für die Endpunkte

  len := 0;
  WHILE (abs(ord(rna[start+len])-ord(rna[ende-len])) in stimmig)  //Solange Farben passend
         and (len<=trunc((ende-start-3)/2)) do inc(len);     // und noch 3 Stellen Abstand
                                                                  //wird len vergrössert
  if len>maxlen then BEGIN
   maxlen := len;                                    //neue längere Harmoniestelle speichern
   maxstart := start;
   maxende := ende;
   END
 END;

if maxlen <> 0 then BEGIN                              //Ausgabe der Längsten Harmoniestelle
 WRITELN('(', maxstart,'-', maxstart+maxlen-1, ',',maxende,'-', maxende-maxlen+1, ')' );
 writeln(copy(rna, maxstart, maxlen));
 writeln(copy(rna, maxende-maxlen+1, maxlen));
 END;

REPEAT UNTIL keypressed;
END.
```

Beispiele:

```
DIE FARBENFOLGE DER RIBONATTER:
GGGAGCGUAGCUCAGUGCGGAGAGCGCCUGCUUUGCACGCAGGAGGUCUGCGGUUCGAUCCCGCGCGCUCCCACCA
(1-7,72-66)
GGGAGCG
CGCUCCC
```

```
DIE FARBENFOLGE DER RIBONATTER:
UUCGAUGUAGCCCCAUUAUAAAAGUCGCUCGUUCGCGCGCCCAAAUGCGCGCUGAAAAUGCGUCUUCU
(35-40,52-47)
GCGCGC
GCGCGC
```

```
DIE FARBENFOLGE DER RIBONATTER:
aaagcugugugguaaacgucguguaaaugaugaucggcguagugaucgucgcgucguaguagaucgugcugcuga
(31-34,64-61)
gauc
gauc
```

Programmablaufprotokoll:

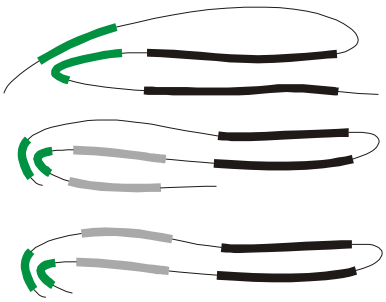
Das Programm ist sehr einfach zu bedienen. Man gibt die gewünschte Farbenfolge ein. Groß- und Kleinschreibung ist egal, hauptsächlich einheitlich. Nach dem Return gibt das Programm die längste Harmoniestelle aus. Nach einem weiteren Tastendruck ist das Programm wieder beendet.

Wie geht dir Ribo-Natter vor um die maximale Farbharmonie herzustellen?

Wenn die Ribo-Natter sich nicht auf eine Ebene beschränken müsste und dazu auch noch dehnbar wie Gummi wäre, hätte sie es leicht. Sie würde sich immer die längste Harmoniestelle suchen und diese Bereiche zusammenlegen. Unter Beachtung, dass die nächsten 3 Ringe von den Harmoniestellen weg nicht verwendet werden dürfen, sucht sie sich wieder die längste Harmoniestelle und legt diese Bereiche wieder zusammen. Das führt sie so lange durch, bis keine Harmoniestellen mehr vorhanden sind. Nach dieser Prozedur wäre aber die Schlange mit hoher Wahrscheinlichkeit nicht mehr eben und Verbindungsstrecken zwischen Harmoniestellen müssten gedehnt oder gestaucht werden. Also muss sich die Ribo-Natter etwas anderes einfallen lassen.
Sie konnte es zum Beispiel auch so machen:

Zu Beginn verschafft sich die Ribo-Natter einen Überblick über alle möglichen Harmoniebereiche und merkt sie sich. Dann beginnt eine rekursive Prozedur bzw. die Natter fängt das Probieren an. Sie wählt sich einen dieser Harmoniebereiche aus und muss Testen, ob sie durch das Zusammenführen dieser Bereiche nicht aus der Ebene heraus muss. Wenn nein, dann ermittelt sie, wie sich die noch übrigen Harmoniebereiche eingeschränkt haben. Wenn dadurch keine weiteren Harmoniebereiche übrig geblieben sind, zählt sie die Anzahl der Harmoniestellen. Wenn die Anzahl größer ist als eine vorherige Variante wird die neue Variante übernommen. Wenn jedoch weitere Harmoniebereiche vorhanden sind, führt sich die Prozedur für jeden dieser Harmoniebereiche noch einmal aus. Das wäre eine ziemlich sichere Methode für die Natter. Sie würde aber ziemlich lang dafür brauchen und am Ende wahrscheinlich schon wieder den Anfang vergessen haben.

Für die Natter wäre eine Methode besser, bei der sie Schritt für Schritt immer Harmoniebereiche bildet und zwar denjenigen der für sie am besten ist. Prinzipiell ist für sie der längste Harmoniebereich am besten. Problematisch dabei ist nur, wenn der längste Harmoniebereich andere kürzere Harmoniebereiche ausschließt, die zusammen größer als er wären. Also müsste sie den Harmonieabschnitt als ersten nehmen, der in Länge*2 – dadurch ausgeschlossene Bereiche am größten ist. Jetzt braucht die Schlange Regeln wie sie einen Ausschluss erkennt.
Durch einen Zusammenschluss sind primär die Stellen der Natter am meisten betroffen, die sich zwischen den zwei zusammengefügten Abschnitten befinden. Im ersten Bild sieht man, dass wen die Natter die schwarzen Bereiche zusammenschließen will der Abstand zwischen den zwei grünen Bereichen ungefähr doppelt so groß sein muss wie der schwarze Harmoniebereich. Wie man im nächsten Beispiel sieht, kann sich aber eine Zusammenkoppelung des schwarzen Bereichs auch die grünen Harmoniestellen auf den zwei Enden auswirken. Die Natter müsste also auch darauf achten. Auffällig ist dabei das die Anzahl der bereits verbundenen Harmoniestellen zwischen den grünen Bereichen ungerade ist. Währe die Anzahl gerade so lässt sich schwerer ein Fall finden in den ein Problem wegen der Länge auftritt. Mein Versuch dies alles in Regeln zu packen sieht so aus:



Davon ausgehen dass ein Bereich als Harmoniebereich zusammengefügt werden sollt.
Alle noch vorhandenen möglichen Harmoniebereiche durchgehen
Zwischen diesen zwei Bereichen die Anzahl der bereits vorhandenen Harmoniebereiche ermitteln
Wenn Anzahl ungerade ist:
Wenn die Anzahl der noch im Harmoniezustand befindenden Farbringe zwischen den Harmoniestellen kleiner ist als die Länge des nächsten Harmonieabschnittes zur Mitte hin.
Entstandener Verlust um die Länge der verlorenen Harmoniestelle erhöhen.
Differenz zwischen im Gedanken zusammengehängter Harmoniestelle und Verlust bilden.
Suche nach einer Harmoniestelle die größer ist als diese Differenz
Wenn vorhanden den selben Vergleich durchführen und wenn die andere Harmoniestelle günstiger ist diese zum weiteren Vergleichen heranziehen.
Wenn dann nur noch im kleinere Harmoniestellen vorhanden sind die Harmonie herstellen und die ganze Prozedur so lange durchführen bis keine möglichen Harmoniestellen mehr vorhanden sind.

Ich denke das bei dieser starken Reduktion des Problems einige Sonderfälle nicht beachtet sind. So wäre es aber für die Schlange sehr einfach.