Abgabetermin: Mi in der KW 14

☐ **Gr. 1,** Dr. G. Kronberger **Name** _____ **Aufwand in h** _____
☐ **Gr. 2,** Dr. H. Gruber
☐ **Gr. 3,** Dr. D. Auer **Punkte** _____ **Kurzzeichen Tutor / Übungsleiter** _____ / _____

1. *m*-Ketten-Problem (4 + 6 Punkte)

a) *Definition*: Eine Zeichenkette ist eine *m*-Kette, wenn sie höchstens *m* unterschiedliche Zeichen enthält.

Beispiele: Die drei Zeichenketten *a*, *ab* und *abcbaac* sind 3-Ketten, die Zeichenkette *abcd* ist aber keine 3-Kette mehr, sondern eine 4-Kette. Eine Zeichenkette *s* der Länge *n* ist natürlich eine *n*-Kette, von Interesse ist aber das kleinste *m* für welches die Bedingung aus der Definition oben für *s* noch gilt. Entwickeln Sie daher eine möglichst effiziente Funktion

```
FUNCTION MinM(s: STRING): INTEGER;
```

zur Ermittlung des kleinsten *m* für eine Zeichenkette *s*.

b) Gegeben sei eine nichtleere Zeichenkette *s* und eine ganze Zahl *m* mit $1 \leq m \leq Length(s)$. Entwickeln Sie eine möglichst effiziente Funktion

```
FUNCTION MaxMStringLen(s: STRING, m: INTEGER): INTEGER;
```

welche die Länge der längsten *m*-Kette, die als Teilkette in *s* enthalten ist, liefert.

2. *Wildcard Pattern Matching* (4 + (5 + 5) Punkte)

Viele Programme, z. B. Texteditoren und Kommandozeilen-Interpretierer diverser Betriebssysteme (engl. *shells*), verwenden eine spezielle Variante der Zeichenkettensuche, die in der Musterkette Platzhalter (auch Jokerzeichen, engl. *wildcards*, genannt) zulässt. Denken Sie z. B. an den MS-DOS/Windows-Befehl *del* *.* bzw. an das äquivalente UNIX-Kommando *rm* *. Hier muss festgestellt werden, ob die Musterkette (*.* bzw. *) zu einem Dateinamen im aktuellen Verzeichnis passt.

Jokerzeichen dürfen nur in Musterketten vorkommen: Dabei steht das Jokerzeichen ? für *ein* beliebiges Zeichen in der Zeichenkette und das Jokerzeichen * für eine *beliebige Anzahl* (null oder mehr) beliebiger Zeichen in der Zeichenkette. Jokerzeichen können auch gemischt und mehrfach in einer Musterkette vorkommen.

Nehmen Sie an, dass sowohl die Muster- als auch die Zeichenkette durch das spezielle Endezeichen $ abgeschlossen ist, welches innerhalb der Ketten nicht vorkommt. Folgende Tabelle zeigt einige einfache *Beispiele*:

Musterkette *p*	**Zeichenkette *s***	***p* und *s* passen zusammen?**
ABC$	ABC$	ja
ABC$	AB$	nein
ABC$	ABCD$	nein
A?C$	AXC$	ja
*$	*beliebige auch leere Kette*	ja
A*C$	AC$	ja
A*C$	AXYZC$	ja

a) Erweitern/ändern Sie den *BruteForce*-Algorithmus für die Zeichenkettensuche so, dass er obige Aufgabenstellung bewältigt, jedoch als Jokerzeichen nur ? (auch mehrfach) in der Musterkette vorkommen darf.

b) Die zusätzliche Behandlung des Jokerzeichens * ist mit den Standard-Algorithmen leider nicht mehr so einfach möglich. Allerdings lässt sich das Problem relativ einfach mittels Rekursion lösen:

   1. Definieren Sie zuerst ein rekursives Prädikat *Matching*(*p*, *s*), das *true* liefert, wenn *p* und *s* zusammenpassen, sonst *false*. Zerlegen Sie dabei sowohl *p* als auch *s* "geschickt" in zwei Teile: in das erste Zeichen und den Rest der Kette.
   2. Implementieren Sie das Prädikat *Matching* in Form einer rekursiven Funktion und testen Sie diese ausführlich.

Für besonders Interessierte: Implementieren Sie eine iterative Variante des Prädikats *Matching* aus 2.b.1, testen Sie diese ausführlich und stellen Sie Laufzeitvergliche der rekursiven und iterativen Variante für lange Eingabeketten an. Als „**Belohnung**“ gibt es bis zu vier Zusatzpunkte.