

Gr. 1, Dr. D. Auer  
 Gr. 2, Dr. G. Kronberger  
 Gr. 3, Dr. S. Wagner

Name Klemens Danner Aufwand in h 8

Punkte \_\_\_\_\_ Kurzzeichen Tutor\*in / Übungsleiter\*in \_\_\_\_\_ / \_\_\_\_\_

### 1. Maximum von zwei oder drei Werten (2 + 2 Punkte)

- a) Implementieren Sie eine Funktion Max2, die das Maximum zweier ganzer Zahlen als Funktionsergebnis liefert.
- b) Implementieren Sie eine Funktion Max3, die das Maximum dreier ganzer Zahlen liefert und dazu die Funktion Max2 möglichst geschickt nutzt.

### 2. Paare befreundeter Zahlen (8 Punkte)

Entwickeln Sie eine Pascal-Funktion, die prüft, ob zwei ganze Zahlen ein *Paar befreundeter Zahlen* (siehe <https://oeis.org/A259180>) sind. Zwei verschiedene Zahlen heißen *befreundet*, wenn jede Zahl gleich der Summe der echten Teiler (einschließlich des Teilers 1, aber ohne die Zahl selbst) der jeweils anderen Zahl ist.

Entwickeln Sie dazu zusätzlich eine Pascal-Funktion, die die Summe der echten Teiler einer gegebenen Zahl berechnet.

*Hinweis:* das kleinste Paar befreundeter Zahlen ist 220 und 284:

Summe der echten Teiler von 220:  $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

Summe der echten Teiler von 284:  $1 + 2 + 4 + 71 + 142 = 220$

Testen Sie Ihre Funktionen mit einem Pascal-Programm, das zwei ganze Zahlen einliest, die Funktion zur Prüfung auf befreundete Zahlen aufruft und das Ergebnis ausgibt.

### 3. Erzeugen von Balkendiagrammen (12 Punkte)

Entwickeln Sie ein Pascal-Programm, das ein druckbares Zeichen und fünf ganze Zahlen (jede im Bereich von 1 bis 10) von der Tastatur einliest. Mit diesen Werten muss dann die von Ihnen zu implementierende

```
PROCEDURE BarChart(ch: CHAR; n1, n2, n3, n4, n5: INTEGER);
```

aufgerufen werden. Diese Prozedur zeigt für jede der fünf Zahlen  $n_i$  (für  $i = 1$  bis 5) einen vertikalen Balken mit der entsprechenden Anzahl  $n_i$  von  $ch$ -Zeichen an. Leere Zeilen, wie im folgenden Beispiel die Zeilen 8 bis 10, dürfen nicht angezeigt werden.

*Beispiel:*

```
ch: X
ni: 3 5 7 4 2
```

7	X
6	X
5	X X
4	X X X
3	X X X X
2	X X X X X
1	X X X X X
-----	
	1 2 3 4 5

**Hinweise:**

1. Geben Sie für alle Ihre Lösungen immer eine „Lösungsidee“ an.
2. Dokumentieren und kommentieren Sie Ihre Pascal-Programme.
3. Geben Sie immer auch Testfälle ab, an denen man erkennen kann, dass Ihr Pascal-Programm funktioniert, und dass es auch in Fehlersituationen entsprechend reagiert.

# **1 Maximum von zwei oder drei Werten**

## **1.1 Lösungsidee**

- a) 2 Werte werden eingelesen. Die Funktion Max2 prüft mittels if-Abfrage welcher Wert größer ist und gibt diesen zurück.
- b) 3 Werte werden eingelesen. Die Funktion Max3 ermittelt das Maximum, indem sie mit der Funktion Max2 aus Aufgabe a) das Maximum der ersten beiden Zahlen ermittelt dann das Maximum aus dem Rückgabewert und der dritten Zahl bestimmt, welches insgesamt das Maximum von allen drei Zahlen ist.

## 1.2 Quellcode

```
1  PROGRAM Maximum;
2
3  // Aufgabe a)
4  FUNCTION Max2(a, b: INTEGER): INTEGER;
5      var maximum: INTEGER;
6
7      BEGIN
8          IF a > b then
9              maximum := a
10         ELSE
11             maximum := b;
12         Max2 := maximum
13     END;
14
15 // Aufgabe b)
16 FUNCTION Max3(a, b, c: INTEGER): INTEGER;
17 BEGIN
18     Max3 := Max2(Max2(a,b),c);
19 END;
20
21 VAR
22     a, b, c: INTEGER;
23
24 BEGIN
25
26     Write('Geben Sie drei Zahlen ein. Das Maximum der ersten beiden und aller
27     drei Zahlen wird ermittelt: ');
28     WriteLn; Read(a,b,c);
29
30     WriteLn('Maximum der ersten beiden Zahlen: ', Max2(a,b));
31     WriteLn('Maximum aller Zahlen: ', Max3(a,b,c));
32 END.
```

Eingabe	Beschreibung	Ausgabe
6,2,4	unterschiedliche pos. Zahlen	Geben Sie drei Zahlen ein. Das Maximum der ersten beiden und aller drei Zahlen wird ermittelt: 6 2 4 Maximum der ersten beiden Zahlen: 6 Maximum aller Zahlen: 6
0,0,0	0 und nur eine Zahl	Geben Sie drei Zahlen ein. Das Maximum der ersten beiden und aller drei Zahlen wird ermittelt: 0 0 0 Maximum der ersten beiden Zahlen: 0 Maximum aller Zahlen: 0
-5,0,1	negative Zahl dabei	Geben Sie drei Zahlen ein. Das Maximum der ersten beiden und aller drei Zahlen wird ermittelt: -5 0 1 Maximum der ersten beiden Zahlen: 0 Maximum aller Zahlen: 1
-15, -21, -1	nur negative Zahlen	Geben Sie drei Zahlen ein. Das Maximum der ersten beiden und aller drei Zahlen wird ermittelt: -15 -21 -1 Maximum der ersten beiden Zahlen: -15 Maximum aller Zahlen: -1

## **2 Befreundete Zahlen**

### **2.1 Lösungsidee**

Zuerst wird eine Funktion implementiert, welche die Summe der echten Teiler zurückgibt. In einer zweiten Funktion wird unter der Verwendung der ersten Funktion geprüft, ob die Summe der Teiler gleich der jeweils anderen Zahl ist und anschließend ein Wahrheitswert zurückgegeben.

Mit for Schleife von der Zahl bis 1 runterzählen. Immer wenn Rest 0, dann den Index zu einer Summenvariable hinzufügen.

## 2.2 Quellcode

```
1  PROGRAM BefreundeteZahlen1;
2  VAR a,b: INTEGER;
3
4  FUNCTION TeilerSumme(a: INTEGER): INTEGER;
5    var Summe, i: INTEGER;
6
7  BEGIN
8    Summe := 0; //init
9    for i := (a - 1) downto 1 do
10      BEGIN
11        IF (a mod i = 0) then
12          Summe := Summe + i;
13      END;
14
15      Teilersumme := Summe;
16  END;
17
18  FUNCTION BefreundeteZahlen(a,b: INTEGER): BOOLEAN;
19
20  BEGIN
21    IF (a<1) or (b<1) then //negative Zahlen machen keinen Sinn, und auch 0
sind keine befreundeten Zahlen
22    BefreundeteZahlen := false
23    ELSE
24    BefreundeteZahlen := (a = TeilerSumme(b)) and (b = TeilerSumme(a));
25  END;
26
27  BEGIN
28    Write('Zwei Zahlen eingeben: '); Read(a,b);
29
30    IF BefreundeteZahlen(a,b) then
31      WriteLn('Diese Zahlen sind befreundet.')
32    ELSE
33      WriteLn('Diese Zahlen sind nicht befreundet.');
34  END.
```

## 2.3 Tests

Eingaben	Beschreibung	Ausgaben
220,284	allgemein	Zwei Zahlen eingeben: 284 220 Diese Zahlen sind befreundet.
1210,1184	allgemein	Zwei Zahlen eingeben: 1210 1184 Diese Zahlen sind befreundet.
100,100	nicht befreundet	Zwei Zahlen eingeben: 100 100 Diese Zahlen sind nicht befreundet.
0,0	nicht befreundet, untere Grenze	Zwei Zahlen eingeben: 0 0 Diese Zahlen sind nicht befreundet.
-220,-284	negativ, nicht befreundet	Zwei Zahlen eingeben: -220 -284 Diese Zahlen sind nicht befreundet.

# **3 Balkendiagramm**

## **3.1 Lösungsidee**

Es werden ein Char und 5 Zahlen eingelesen, welche von der Prozedur übernommen werden. Wenn die eingelesenen Zahlen im Wertebereich liegen, fährt das Programm fort. Mit der Funktion Max2 aus Aufgabe 1 wird von den eingelesenen Zahlen das Maximum ermittelt, um die Anzahl der Zeilen herauszufinden.

Für jede Zeile wird für alle Werte  $n_i$  mittels einer anderen Prozedur geprüft, ob der Index der aktuellen Zeile kleiner oder gleich des jeweiligen Wertes  $n_i$  ist. Wenn sie kleiner ist, wird der Char geschrieben, ansonsten ein Leerzeichen.

## 3.2 Quellcode

```
1 PROGRAM Balkendiagramm;
2
3 VAR
4     n1, n2, n3, n4, n5: INTEGER;
5     ch1: string; //um runtime error bei Falscheingabe zu vermeiden
6
7
8 //Funktion zur Berechnung vom Max
9 FUNCTION Max2(a, b: INTEGER): INTEGER;
10    var maximum: INTEGER;
11    BEGIN
12        IF a > b then
13            maximum := a
14        ELSE
15            maximum := b;
16    Max2 := maximum
17    END;
18
19 //Entscheidet für eine Stelle, ob ein ch oder ' ' geschrieben wird
20 PROCEDURE SetCharOrSpace(ch: CHAR; zeile, ni: INTEGER);
21
22    BEGIN
23
24        IF zeile <= ni then
25            Write(' ', ch)
26        ELSE
27            Write(' ');
28
29    END;
30
31
32 PROCEDURE BarChart(ch: CHAR; n1, n2, n3, n4, n5: INTEGER);
33 //Block
34 VAR
35     nMax: INTEGER;
36     j: INTEGER;
37     BEGIN
38
39         //BarChart ausgeben, wenn im Wertebereich
40         IF (n1 in [1..10]) and (n2 in [1..10]) and (n3 in [1..10]) and (n4 in
41 [1..10]) and (n5 in [1..10]) then
42             BEGIN
43                 //Maximum für Anzahl der Zeilen ermitteln
```

```

43     nMax := Max2(Max2(Max2(n1, n2), n3), n4), n5);
44
45
46     for j := nMax downto 1 do
47         BEGIN
48             //Zeile basteln und ausgeben, in eigene Prozedur auslagern
49             Write(j:2, '|');
50             SetCharOrSpace(ch, j, n1);
51             SetCharOrSpace(ch, j, n2);
52             SetCharOrSpace(ch, j, n3);
53             SetCharOrSpace(ch, j, n4);
54             SetCharOrSpace(ch, j, n5);
55
56             WriteLn;
57         END;
58
59         WriteLn(' +-----');
60         WriteLn('    1 2 3 4 5');
61     END
62 ELSE
63     WriteLn('Der Wertebereich ist 1 bis 10')
64 END;
65
66 BEGIN
67     WriteLn('Geben Sie einen char und 5 Zahlen zwischen 1 und 10 ein, um ein
68 Balkendiagramm zu erstellen!');
69     WriteLn;
70     Read(ch1, n1, n2, n3, n4, n5);
71
72     if length(ch1) = 1 then
73         BEGIN
74             WriteLn;
75             BarChart(ch1[1], n1, n2, n3, n4, n5);
76         END
77     ELSE
78         WriteLn('Bitte einen Char eingeben!');
79     END.

```

### 3.3 Tests

Eingaben	Begründung	Ausgaben
X,3,5,7,4,2	Aufgabe	<pre> 7       X 6       X 5     X X 4    X X X 3  X X X X 2  X X X X X 1  X X X X X +----- 1 2 3 4 5 </pre>
O,2,5,10,4,8	maximaler Wert, anderer Char	<pre> 10      0 9      0 8      0  0 7      0  0 6      0  0 5    0 0  0 4    0 0 0 0 3    0 0 0 0 2  0 0 0 0 0 1  0 0 0 0 0 +----- 1 2 3 4 5 </pre>
A,1,1,1,1,1	minimaler Wert	<pre> 1  A A A A A +----- 1 2 3 4 5 </pre>
B,1,4,0,6,5	außerhalb von Wertebereich	<p>B 1 4 0 6 5</p> <p>Der Wertebereich ist 1 bis 10</p>
C,1,4,11,6,5	außerhalb von Wertebereich	<p>C 1 4 11 6 5</p> <p>Der Wertebereich ist 1 bis 10</p>
AB,1,2,3,4,5	String eingegeben	<p>AB 1 2 3 4 5</p> <p>Bitte einen Char eingeben!</p>