

<input type="checkbox"/> Gr. 1, Dr. D. Auer	Name _____	Aufwand in h _____
<input type="checkbox"/> Gr. 2, Dr. G. Kronberger		
<input type="checkbox"/> Gr. 3, Dr. S. Wagner	Punkte _____	Kurzzeichen Tutor / Übungsleiter _____ / _____

1. Feldverarbeitung mit offenen Feldparametern**(8 Punkte)**

Gegeben sind zwei beliebig große Felder $a1$ und $a2$, die $n1$ bzw. $n2$ positive ganze Zahlen in aufsteigend sortierter Reihenfolge enthalten. Gesucht ist eine Pascal-Prozedur

```
PROCEDURE Merge (a1: ARRAY OF INTEGER; n1: INTEGER;
                 a2: ARRAY OF INTEGER; n2: INTEGER;
                 VAR a3: ARRAY OF INTEGER; VAR n3: INTEGER);
```

die das Feld $a3$ so belegt, dass darin alle $n3$ Zahlen in aufsteigend sortierter Reihenfolge enthalten sind, die in $a1$ und in $a2$ vorkommen. Beachten Sie, dass in jedem Feld (also auch im Ergebnisfeld $a3$) Werte mehrfach vorkommen können. Im Fehlerfall (Überlauf in $a3$) soll $n3$ auf -1 gestellt werden.

Beispiel:

$a1 =$	2	4	4	10	15	15	...	$n1 = 6$				
$a2 =$	3	4	5	10	...			$n2 = 4$				
<hr/>												
$a3 =$	2	3	4	4	4	5	10	10	15	15	und	$n3 = 10$

2. Durchschnittlicher Strombedarf**(8 Punkte)**

Gegeben ist ein beliebig großes Feld *measuredValues*, das n Messwerte des täglichen Strombedarfs (z.B. in der Einheit kWh) eines Haushalts enthält. Gesucht ist eine Pascal-Prozedur

```
PROCEDURE AveragePerWeekday (measuredValues: ARRAY OF REAL; n: INTEGER;
                             VAR avg: WeekdayValues);
```

die den durchschnittlichen Strombedarf für jeden Wochentag ermittelt und in das Feld *avg* speichert. Gehen Sie davon aus, dass das erste (achte, ...) Feldelement von *measuredValues* den Wert für einen Montag speichert, das zweite (neunte, ...) den Wert für einen Dienstag usw. (siehe Beispiel).

Gehen Sie außerdem von folgenden Deklarationen aus:

```
CONST
    weekdays = 7;
TYPE
    WeekdayValues = ARRAY[1..weekdays] OF REAL;
```

Beispiel:

	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	...
$mv =$	8.0	8.0	21.0	12.0	8.5	9.5	11.0	8.0	9.0	9.0	8.0	... $n = 11$
<hr/>												
$avg =$	8.0	8.5	15.0	10.0	8.5	9.5	11.0					

3. Buchstabensalat → Bcushatebsnlaat

(8 Punkte)

Vielleicht haben Sie schon einmal davon gehört, dass man Wörter auch dann lesen kann, wenn die Reihenfolge der Buchstaben beliebig vertauscht werden, der erste und letzte Buchstabe aber unverändert bleibt.

Entwickeln Sie ein Pascal-Programm, das eine Zeichenkette s einliest, die Buchstaben der in s enthaltenen Wörter umordnet und anschließend die Zeichenkette s wieder ausgibt.

Wählen Sie eine geeignete Strategie für das Umordnen der Buchstaben, z.B. durch Vertauschen von jeweils zwei benachbarten Buchstaben (siehe Beispiele) oder durch Umkehren der Buchstabenreihenfolge. Beachten Sie aber, dass der erste und letzte Buchstabe eines Wortes nicht verändert werden darf.

Beispiel 1:

Eingabe: Software ist der Geist in der Maschine.

Ausgabe: Sfowtrae ist der Giest in der Msahcnie.

Beispiel 2:

Eingabe: Haensel und Gretel verliefen sich im Wald.

Ausgabe: Heasnel und Geretl vreilfeen scih im Wlad.

Hinweise:

1. Geben Sie für alle Ihre Lösungen immer eine „Lösungsidee“ an.
2. Dokumentieren und kommentieren Sie Ihre Algorithmen.
3. Bei Programmen: Geben Sie immer auch Testfälle ab, an denen man erkennen kann, dass Ihr Programm funktioniert, und dass es auch in Fehlersituation entsprechend reagiert.

1)

Lösungsidee:

Erstmal muss das outputarray(a3) mit den kleinsten Zahlen von a1 und a2(sortiert) gefüllt werden bis n1 oder n2 erreicht wurde. Je nachdem welches Feld(a1 oder a2) als erstes „verbraucht“ wurde müssen dann die Zahlen vom anderen Feld im outputarray(a3) eingefüllt werden.

Quelltext

```
PROGRAM Feldbearbeitung;
```

```
TYPE
```

```
intarr = ARRAY[1..100] OF INTEGER;
```

```
PROCEDURE Initarr(n : INTEGER; VAR arr : ARRAY OF INTEGER);
```

```
VAR
```

```
i : INTEGER;
```

```
BEGIN
```

```
    i := LOW(arr);
```

```
    WHILE i <> n DO BEGIN
```

```
        Read(arr[i]);
```

```
        Inc(i);
```

```
    END;
```

```
END;
```

```
PROCEDURE Merge(a1:ARRAY OF INTEGER;n1: INTEGER;a2: ARRAY OF INTEGER; n2: INTEGER;VAR a3:  
    ARRAY OF INTEGER; VAR n3:INTEGER);
```

```
VAR
```

```
i,j,k : INTEGER;
```

```
BEGIN
```

```
    n3 := n1 + n2;
```

```
    i:=0;
```

```
    j:=0;
```

```
    k:=0;
```

```
    WHILE (i < n1) AND (j < n2) DO BEGIN
```

```
        IF a1[i] <= a2[j] THEN BEGIN
```

```
            a3[k] := a1[i];
```

```
            Inc(k);
```

```
            Inc(i);
```

```
        END
```

```
        ELSE BEGIN
```

```
            a3[k] := a2[j];
```

```
            Inc(k);
```

```
            Inc(j);
```

```
        END;
```

```
    END;
```

```
    WHILE i < n1 DO BEGIN
```

```
        a3[k] := a1[i];
```

```
        Inc(i);
```

```
        Inc(k);
```

```

    END;
    WHILE j < n2 DO BEGIN
        a3[k] := a2[j];
        Inc(j);
        Inc(k);
    END;
END;

PROCEDURE WriteIntArray(yourArray : ARRAY OF INTEGER; n : INTEGER);
VAR
    i : INTEGER;
BEGIN
    FOR i := LOW(yourArray) TO n-1 DO BEGIN
        Write(yourArray[i], ' ');
    END;
END;

VAR
    a1,a2,a3 :intarr;
    n1,n2,n3 :INTEGER;
BEGIN
    WriteLn('wie gross ist das erste Feld?');
    Read(n1);
    WriteLn('Zahlen fuers erste Feld eingeben');
    Initarr(n1,a1);
    WriteLn('wie gross ist das zweite Feld?');
    Read(n2);
    WriteLn('Zahlen fuers zweite Feld eingeben');
    Initarr(n2,a2);
    Merge(a1,n1,a2,n2,a3,n3);
    WriteIntArray(a3,n3);
END.

```

Testfall1)

```

wie gross ist das erste Feld?
1
Zahlen fuers erste Feld eingeben
1
wie gross ist das zweite Feld?
1
Zahlen fuers zweite Feld eingeben
1
output:
1 1

```

Testfall2)

```
wie gross ist das erste Feld?
6
Zahlen fuers erste Feld eingeben
2 4 4 10 15 15
wie gross ist das zweite Feld?
4
Zahlen fuers zweite Feld eingeben
3 4 5 10
output:
2 3 4 4 4 5 10 10 15 15 █
```

Testfall3)

```
wie gross ist das erste Feld?
10
Zahlen fuers erste Feld eingeben
1 2 3 3 3 4 7 8 9 10
wie gross ist das zweite Feld?
5
Zahlen fuers zweite Feld eingeben
3 3 10 11 12
output:
1 2 3 3 3 3 3 4 7 8 9 10 10 11 12 █
```

2)

Lösungsidee:

Ich habe die Aufgabe mithilfe einer For und einer While Schleife gelöst. Die For Schleife zählt bis zum Sonntag(7) während die While Schleife für jeden Tag den Durchschnitt ausrechnet. Die While Schleife soll solange laufen bis der Index von der For Schleife + 7 größer gleich „n“ wird.

Quelltext:

```
PROGRAM AvgPower;

CONST
    weekdays = 7;

TYPE
    WeekdayValues = ARRAY[1..weekdays] OF REAL;
    measuredValues = ARRAY[1..100] OF REAL;

PROCEDURE InitArray(VAR measuredValues : ARRAY OF REAL;VAR n : INTEGER);
VAR
    input : REAL;
BEGIN
```

```
Read(input);
n := LOW(measuredValues);
WHILE input <> 0 DO BEGIN
    measuredValues[n] := input;
    Read(input);
    Inc(n);
END;
Dec(n);
END;
```

```
PROCEDURE AveragePerWeekday(measuredValues:ARRAY OF REAL;n: INTEGER;VAR avg: WeekdayValue
s);
VAR
sum : REAL;
week,count : INTEGER;
i :INTEGER;
BEGIN
    FOR i:= LOW(measuredValues) TO weekdays-1 DO BEGIN
        sum := measuredValues[i];
        count := 1;
        week := weekdays;
        WHILE i+week <= n DO BEGIN
            sum := sum + measuredValues[i+week];
            Inc(count);
            week := week + weekdays;
        END;
        avg[i+1] := sum/count;
    END;
END;
```

```
VAR
n,i :INTEGER;
inputValues : measuredValues;
avg : WeekdayValues;
BEGIN
    WriteLn('Bitte Zahlen eingeben');
    InitArray(inputValues,n);
    AveragePerWeekday(inputValues,n,avg);
    WriteLn('    MO    DI    MI    DO    FR    SA    SO');
    FOR i:= 1 TO weekdays DO BEGIN
        Write(avg[i]:7:3);
    END;
END.
```

Testfall1)

```
Bitte Zahlen eingeben
8.0 8.0 21.0 12.0 8.5 9.5 11.0 8.0 9.0 9.0 8.0 0
  MO    DI    MI    DO    FR    SA    SO
 8.000  8.500 15.000 10.000  8.500  9.500 11.000
```

Testfall2)

```
Bitte Zahlen eingeben
1
0
  MO    DI    MI    DO    FR    SA    SO
 1.000  0.000  0.000  0.000  0.000  0.000  0.000
```

3)

Lösungsidee:

Mit der Aufgabe hatte ich ein paar Schwierigkeiten. Am Anfang wollte ich den String direkt bearbeiten, habe aber nicht wirklich hinbekommen. Ich habe es jetzt gelöst, indem ich die Wörter von einem String aufspalte und in ein String array speichere. Dann wird anschließend jedes Wort im array verarbeitet. Hierbei wird jeder Buchstabe mit dem Nebenbuchstaben vertauscht wobei der Erste und letzte Buchstabe ignoriert wird.

Quelltext:

```
PROGRAM Buchstabesalat;

TYPE
stringArr = ARRAY[1..100] OF String;

PROCEDURE removeSpace(VAR inputWords : ARRAY OF STRING);
VAR
i,j : INTEGER;
space : CHAR;
BEGIN
  space := CHAR(32);
  FOR i:= LOW(inputWords) TO HIGH(inputWords) DO BEGIN
    FOR j:= 1 TO Length(inputWords[i]) DO BEGIN
      IF inputWords[i][j] = space THEN
        DELETE(inputWords[i],j,1);
    END;
  END;
END;
```

```
PROCEDURE splitInput (input : STRING; VAR inputWords : stringArr);
VAR
i,pos1,pos2,j,NextWord : INTEGER;
space : CHAR;
BEGIN
space := CHAR(32);
pos1 := 1;
NextWord := 1;
FOR i := 1 TO Length(input) DO BEGIN
    IF (input[i] = space) OR (i = Length(input)) THEN BEGIN
        pos2 := i;
        FOR j := pos1 TO pos2 DO BEGIN
            inputWords[NextWord] := Concat(inputWords[NextWord],input[j]);
        END; //FOR
        pos1 := pos2 +1;
        Inc(NextWord);
    END; //IF
END; //FOR
removeSpace(inputWords);
END; //splitInput

PROCEDURE WriteStrArray(yourArray : ARRAY OF STRING; Length : INTEGER);
VAR
i : INTEGER;
BEGIN
FOR i := LOW(yourArray) TO HIGH(yourArray) DO BEGIN
    Write(yourArray[i], ' ');
END;
END;

PROCEDURE Swap(VAR a,b : CHAR);
VAR c : CHAR;
begin
    c := a;
    a:= b;
    b := c;
end;

PROCEDURE swapChars(VAR inputWords : ARRAY OF STRING);
VAR
i,j : INTEGER;
BEGIN
FOR i:= LOW(inputWords) TO HIGH(inputWords) DO BEGIN
    FOR j := 1 TO Length(inputWords[i]) DO BEGIN
        IF(j <> 1) AND (j <> Length(inputWords[i])) AND (j+1 <> Length(inputWords[i])) THEN
        BEGIN
            swap(inputWords[i][j],inputWords[i][j+1]);
            Inc(j)
        END;
    END;
END;
```



```
END;  
END;  
  
VAR  
inputWords : stringArr;  
input : STRING;  
BEGIN  
WriteLn('Bitte Satz eingeben der gemischt werden soll');  
ReadLn(input);  
splitInput(input,inputWords); //DONE  
swapChars(inputWords);//DONE  
WriteStrArray(inputWords,Length(inputWords));//DONE  
END.
```

Testfall1)

```
Bitte Satz eingeben der gemischt werden soll  
Software ist der Geist in der Maschine.  
Sfowtrae ist der Giest in der Msahcnie.
```

Testfall2)

```
Bitte Satz eingeben der gemischt werden soll  
Haensel und Gretel verliefen sich im Wald.  
Heasnel und Geretl vreilfeen scih im Wlad.
```