# Strukturierte Programmierung Oliver Heil ITB1\_2a

## Aufgabe 1 Sortieren dreier Zahlen

1. Pseudocode
2. sorterV1(↓↑**integer** x, ↓↑**integer** y, ↓↑**integer** z)
3. **integer** biggest
4. **integer** middle
5. **integer** smallest

8. **if** (x > y && x > z) **then**
9. biggest := x
10. **else**
11. **if** (y > x && y > z) **then**
12. biggest := y
13. **else**
14. biggest := z
15. **end**
16. **end**
18. **if** (x < y && x < z) **then**
19. smallest := x
20. **else**
21. **if** (y < x && y < z) **then**
22. smallest := y
23. **else**
24. smallest := z
25. **end**
26. **end**
28. **if** (x < biggest && x > smallest) **then**
29. middle := x
30. **else**
31. **if** (y < biggest && y > smallest) **then**
32. middle := y
33. **else**
34. middle := z
35. **end**
36. **end**
38. x := smallest
39. y := middle
40. z := biggest
42. **end**
43. sorterV2(↓↑**integer** x, ↓↑**integer** y, ↓↑**integer** z)
44. **integer** numbers[3] = {x,y,z}
46. **for** (n=3; n>1; n=n-1)**do**
47. **for** (i=0; i<n-1; i=i+1)**do**
48. **if** (numbers[i] > numbers[i+1])**then**
49. numbers.swap(i, i+1)
50. **end**
51. **end**
52. **end**
54. x := numbers[0]
55. y := numbers[1]
56. z := numbers[2]
57. **end**
58. sorterV3(↓↑**integer** x, ↓↑**integer** y, ↓↑**integer** z)
59. **if**(y > x) **then**
60. Swap(x,y)
61. **end**
63. **if** (z > y) **then**
64. Swap(y,z)
65. **end**
67. **if** (y > x) **then**
68. Swap(x,y)
69. **end**
70. **end**
71. Ablaufdiagramm

SorterV1





Sorter V2



Sorter V3

# Aufgabe 2 Palindrom

1. Palindromtest(↓**string** text, ↑**bool** result)
3. result := **true**
4. **integer** n = text.length() - 1
5. **integer** i = 0
7. **while** (result == **true** **and** i < n) **do**
8. **if**(text[n] != text[i]) **then**
9. result := **false**
10. **end**
12. i := i+1
13. n := n-1
14. **end**
15. **end**

# Aufgabe 3 Häufigkeitsanalyse

1. VowelOccurencies(↓**string** text, ↑**integer** freqA, ↑**integer** freqE, ↑**integer** freqI, ↑**integer** freqO, ↑**integer** freqU)
3. **integer** i = 0
4. **integer** textlength = text.length()
5. **integer** Vowels[5] = {0,0,0,0,0}
7. **while** (i < textlength) **do**
8. **if** (text[i] == 'A' || text[i] == 'a') **then**
9. Vowels[0] = Vowels[0] +1
10. **end**
11. **if** (text[i] == 'E' || text[i] == 'e') **then**
12. Vowels[1] = Vowels[1] +1
13. **end**
14. **if** (text[i] == 'I' || text[i] == 'i') **then**
15. Vowels[2] = Vowels[2] +1
16. **end**
17. **if** (text[i] == 'O' || text[i] == 'o') **then**
18. Vowels[3] = Vowels[3] +1
19. **end**
20. **if** (text[i] == 'U' || text[i] == 'u') **then**
21. Vowels[4] = Vowels[4] +1
22. **end**
23. i = i + 1
24. **end**
26. freqA := (Vowels[0] \* 100) / textlength
27. freqE := (Vowels[1] \* 100) / textlength
28. freqI := (Vowels[2] \* 100) / textlength
29. freqO := (Vowels[3] \* 100) / textlength
30. freqU := (Vowels[4] \* 100) / textlength
32. **end**

text = "Ich bin ein Test" textlength = 16

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | E | I | O | U | i | Überprüfung | Befehl |
| 0 | 0 | 0 | 0 | 0 | 0 | i < textlength | - |
| 0 | 0 | 0 | 0 | 0 | 0 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 0 | Check Vowels | - |
| 0 | 0 | 1 | 0 | 0 | 0 | └ TRUE | Vowels[2] + 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | i < textlength | i + 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | └ TRUE | - |
| 0 | 0 | 1 | 0 | 0 | 1 | Check Vowels | - |
| 0 | 0 | 1 | 0 | 0 | 2 | i < textlength | i + 1 |
| 0 | 0 | 1 | 0 | 0 | 2 | └ TRUE | - |
| 0 | 0 | 1 | 0 | 0 | 2 | Check Vowels | - |
| 0 | 0 | 1 | 0 | 0 | 3 | i < textlength | i + 1 |
| 0 | 0 | 1 | 0 | 0 | 3 | └ TRUE | - |
| 0 | 0 | 1 | 0 | 0 | 3 | Check Vowels | - |
| 0 | 0 | 1 | 0 | 0 | 4 | i < textlength | i + 1 |
| 0 | 0 | 1 | 0 | 0 | 4 | └ TRUE | - |
| 0 | 0 | 1 | 0 | 0 | 4 | Check Vowels | - |
| 0 | 0 | 1 | 0 | 0 | 5 | i < textlength | i + 1 |
| 0 | 0 | 1 | 0 | 0 | 5 | └ TRUE | - |
| 0 | 0 | 1 | 0 | 0 | 5 | Check Vowels | - |
| 0 | 0 | 2 | 0 | 0 | 5 | └ TRUE | Vowels[2] + 1 |
| 0 | 0 | 2 | 0 | 0 | 6 | i < textlength | i + 1 |
| 0 | 0 | 2 | 0 | 0 | 6 | └ TRUE | - |
| 0 | 0 | 2 | 0 | 0 | 6 | Check Vowels | - |
| 0 | 0 | 2 | 0 | 0 | 7 | i < textlength | i + 1 |
| 0 | 0 | 2 | 0 | 0 | 7 | └ TRUE | - |
| 0 | 0 | 2 | 0 | 0 | 7 | Check Vowels | - |
| 0 | 0 | 2 | 0 | 0 | 8 | i < textlength | i + 1 |
| 0 | 0 | 2 | 0 | 0 | 8 | └ TRUE | - |
| 0 | 0 | 2 | 0 | 0 | 8 | Check Vowels | - |
| 0 | 1 | 2 | 0 | 0 | 8 | └ TRUE | Vowels[1] + 1 |
| 0 | 1 | 2 | 0 | 0 | 9 | i < textlength | i + 1 |
| 0 | 1 | 2 | 0 | 0 | 9 | └ TRUE | - |
| 0 | 1 | 2 | 0 | 0 | 9 | Check Vowels | - |
| 0 | 1 | 3 | 0 | 0 | 9 | └ TRUE | Vowels[2] + 1 |
| 0 | 1 | 3 | 0 | 0 | 10 | i < textlength | i + 1 |
| 0 | 1 | 3 | 0 | 0 | 10 | └ TRUE | - |
| 0 | 1 | 3 | 0 | 0 | 10 | Check Vowels | - |
| 0 | 1 | 3 | 0 | 0 | 11 | i < textlength | i + 1 |
| 0 | 1 | 3 | 0 | 0 | 11 | └ TRUE | - |
| 0 | 1 | 3 | 0 | 0 | 11 | Check Vowels | - |
| 0 | 1 | 3 | 0 | 0 | 12 | i < textlength | i + 1 |
| 0 | 1 | 3 | 0 | 0 | 12 | └ TRUE | - |
| 0 | 1 | 3 | 0 | 0 | 12 | Check Vowels | - |
| 0 | 1 | 3 | 0 | 0 | 13 | i < textlength | i + 1 |
| 0 | 1 | 3 | 0 | 0 | 13 | └ TRUE | - |
| 0 | 1 | 3 | 0 | 0 | 13 | Check Vowels | - |
| 0 | 2 | 3 | 0 | 0 | 13 | └ TRUE | Vowels[1] + 1 |
| 0 | 2 | 3 | 0 | 0 | 14 | i < textlength | i + 1 |
| 0 | 2 | 3 | 0 | 0 | 14 | └ TRUE | - |
| 0 | 2 | 3 | 0 | 0 | 14 | Check Vowels | - |
| 0 | 2 | 3 | 0 | 0 | 15 | i < textlength | i + 1 |
| 0 | 2 | 3 | 0 | 0 | 15 | └ TRUE | - |
| 0 | 2 | 3 | 0 | 0 | 15 | Check Vowels | - |
| 0 | 2 | 3 | 0 | 0 | 16 | i < textlength | i + 1 |
| 0 | 2 | 3 | 0 | 0 | 16 | └ FALSE | - |
|  |  |  |  |  |  |  |  |
| freqA | **freqE** | **freqI** | **freqO** | **freqU** | - | - | - |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[0] \* 100) / (textlength) |
| 0 | 12 | 0 | 0 | 0 | - | - | (Vowels[1] \* 100) / (textlength) |
| 0 | 12 | 18 | 0 | 0 | - | - | (Vowels[2] \* 100) / (textlength) |
| 0 | 12 | 18 | 0 | 0 | - | - | (Vowels[3] \* 100) / (textlength) |
| 0 | 12 | 18 | 0 | 0 | - | - | (Vowels[4] \* 100) / (textlength) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | E | I | O | U | i | Überprüfung | Befehl |
| 0 | 0 | 0 | 0 | 0 | 0 | i < textlength | - |
| 0 | 0 | 0 | 0 | 0 | 0 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 0 | Check Vowels | - |
| 1 | 0 | 0 | 0 | 0 | 0 | └ TRUE | Vowels[0] + 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | i < textlength | i + 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | └ TRUE | - |
| 1 | 0 | 0 | 0 | 0 | 1 | Check Vowels | - |
| 2 | 0 | 0 | 0 | 0 | 1 | └ TRUE | Vowels[0] + 1 |
| 2 | 0 | 0 | 0 | 0 | 2 | i < textlength | i + 1 |
| 2 | 0 | 0 | 0 | 0 | 2 | └ TRUE | - |
| 2 | 0 | 0 | 0 | 0 | 2 | Check Vowels | - |
| 2 | 0 | 1 | 0 | 0 | 2 | └ TRUE | Vowels[2] + 1 |
| 2 | 0 | 1 | 0 | 0 | 3 | i < textlength | i + 1 |
| 2 | 0 | 1 | 0 | 0 | 3 | └ TRUE | - |
| 2 | 0 | 1 | 0 | 0 | 3 | Check Vowels | - |
| 2 | 0 | 2 | 0 | 0 | 3 | └ TRUE | Vowels[2] + 1 |
| 2 | 0 | 2 | 0 | 0 | 4 | i < textlength | i + 1 |
| 2 | 0 | 2 | 0 | 0 | 4 | └ FALSE | - |
|  |  |  |  |  |  |  |  |
| freqA | **freqE** | **freqI** | **freqO** | **freqU** | - | - | - |
| 50 | 0 | 0 | 0 | 0 | - | - | (Vowels[0] \* 100) / (textlength) |
| 50 | 0 | 0 | 0 | 0 | - | - | (Vowels[1] \* 100) / (textlength) |
| 50 | 0 | 50 | 0 | 0 | - | - | (Vowels[2] \* 100) / (textlength) |
| 50 | 0 | 50 | 0 | 0 | - | - | (Vowels[3] \* 100) / (textlength) |
| 50 | 0 | 50 | 0 | 0 | - | - | (Vowels[4] \* 100) / (textlength) |

text = "AAII" textlength = 4

text = "GGGG" textlength = 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | E | I | O | U | i | Überprüfung | Befehl |
| 0 | 0 | 0 | 0 | 0 | 0 | i < textlength | - |
| 0 | 0 | 0 | 0 | 0 | 0 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 0 | Check Vowels | - |
| 0 | 0 | 0 | 0 | 0 | 1 | i < textlength | i + 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 1 | Check Vowels | - |
| 0 | 0 | 0 | 0 | 0 | 2 | i < textlength | i + 1 |
| 0 | 0 | 0 | 0 | 0 | 2 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 2 | Check Vowels | - |
| 0 | 0 | 0 | 0 | 0 | 3 | i < textlength | i + 1 |
| 0 | 0 | 0 | 0 | 0 | 3 | └ TRUE | - |
| 0 | 0 | 0 | 0 | 0 | 3 | Check Vowels | - |
| 0 | 0 | 0 | 0 | 0 | 4 | i < textlength | i + 1 |
| 0 | 0 | 0 | 0 | 0 | 4 | └ FALSE | - |
|  |  |  |  |  |  |  |  |
| freqA | **freqE** | **freqI** | **freqO** | **freqU** | - | - | - |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[0] \* 100) / (textlength) |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[1] \* 100) / (textlength) |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[2] \* 100) / (textlength) |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[3] \* 100) / (textlength) |
| 0 | 0 | 0 | 0 | 0 | - | - | (Vowels[4] \* 100) / (textlength) |