Dr. B. Güsmann VGU

Electrical Engineering and Information Technology, B.Eng.
Introduction to the C Programming Language: Exercises

Exercise Sheet 8

---

1. Develop a function statistic() to compute some statistical values. The function has 4 parameters. The first is an array with 10 float-values, which is the import to the function. With the second parameter, the arithmetic mean

$$m = \frac{a[0] + a[1] + \ldots + a[9]}{10}$$

shall be exported from the function. With the third parameter, the variance

$$v = \frac{(a[0] - m)^2 + (a[1] - m)^2 + \ldots + (a[9] - m)^2}{10}$$

shall be exported from the function. With the fourth parameter, the standard deviation

$$s = \sqrt{v}$$

shall be exported from the function. Find out how to use sqrt() to compute the standard deviation. Do you need an include file?

Use pointers to export the values. Write a main() to test your function.

2. Caveats using scanf()

a)
```
char text[40];
scanf("%s", text);
printf("\n%s\n", text);
```

We can enter a line of text (without blanks). The input is finished with <enter/return > Finally the line is printed onto the screen.

b) Read a char value, a float value, an int value and a text:

```
k<enter/return>
5.6<enter/return>
345<enter/return>
Lecture in Vietnam<enter/return>
```

If you read 345 with scanf() and the following text with getchar() and a while-loop, then you have the problem, that scanf() reads 345 and lets the following ‘\n’ in the keyboard buffer. So, the first character which you get using getchar() and a while-loop, is a ‘\n’, which ends immediately your reading. Corrective action is an additional getchar() after scanf().

If scanf() follows a scanf(), the old ‘\n’ is no problem because the second scanf() reads ‘\n’ out of the keyboard buffer and discards it.