

Programmieren in C, Testat 5

1. Schreiben Sie ein Programm, das ein Zeichen c , ein Wort w der Länge ≤ 20 und zwei ganze Zahlen m, n von der Tastatur einliest und dann automatisch folgende Fragen mit 1 für “ja” und 0 für “nein” richtig beantwortet.

- a) Ist c ein Großbuchstabe ?
- b) Liegt m in $[1, 20]$ und ist c in diesem Fall das m -te Zeichen von w ?
- c) Ist m durch n oder ist n durch m teilbar ?
- d) Gilt $m = n$ und $n^2 < 17$?
- e) Ist eine der beiden Ungleichungen $m - n > m + n$ und $m \cdot n \geq -m \cdot n$ falsch ?
- f) Hat die Fließkommaauswertung von m/n die erste Nachkommastelle 2 ?

Bestimmen Sie die jeweilige Antwort $A \in \{0, 1\}$ ausschließlich mit arithmetischen und logischen Operatoren aus Kapitel 3 des Skriptes und geben Sie diese auf dem Bildschirm aus.¹ Eine mögliche Ein-/Ausgabe sieht folgendermaßen aus:

```
Geben Sie ein Zeichen ein: c = u
Geben Sie ein Wort ein (<= 20 Zeichen): w = Haus
Geben Sie eine ganze Zahl ein: m = 3
... und noch eine: n = -9
```

```
Ist 'u' ein Großbuchstabe?
Antwort: 0
```

```
Liegt 3 in [1,20] und ist 'u' in diesem Fall das 3-te Zeichen von 'Haus' ?
Antwort: 1
```

```
Ist 3 durch -9 oder ist -9 durch 3 teilbar ?
Antwort: 1
```

```
Gilt 3 = -9 und -9*-9 < 17 ?
Antwort: 0
```

```
Ist eine der beiden Ungleichungen 3 - -9 > 3 + -9 und 3*-9 >= -3*-9 falsch ?
Antwort: 1
```

```
Hat die Fließkommaauswertung von 3/-9 = -0.33... die erste Nachkommastelle 2 ?
Antwort: 0
```

Finden Sie Werte m und n , sodass e) mit “nein” und f) mit “ja” beantwortet wird.

¹Für f) dürfen Sie auch explizites Typecasting nutzen, müssen es aber nicht.

2. Im folgenden Programm fehlen einige arithmetische Operationen oder sind unvollständig (s. Skript 3.1). Vervollständigen Sie das Programm an den durch Leerzeichen _ gekennzeichneten Stellen derart, dass es nachfolgende Ausgabe erzeugt. Erklären Sie Ihre Ergänzungen ganz genau!

```
1  #include <stdio.h>
2  int main(void){
3      int x=27;
4      float y=-1.5;
5      char c1='U', c2;
6      char str[]="String",*s1=str,*s2;
7      y__; printf("\n y = %f",y);
8      x _= 7; printf("\n x = %d",x);
9      x *= 6; printf("\n %d",x _ 13);
10     c2 = c1__; printf("\n c1 = %c, c2 = %c",c1,c2);
11     s2 = s1__; printf("\n s1 = %s, s2 = %s ",s1,s2);
12     *s1 = *s2__; printf("\n *s1 = %c, *s2 = %c \n",*s1,*s2);
13     return 0;
14 }
```

Ausgabe: y = -0.500000
x = 6
2
c1 = T, c2 = U
s1 = tring, s2 = String
*s1 = S, *s2 = S

English translation

1. Write a program that reads a character c , a word w of length ≤ 20 and two integers m, n from the keyboard and that automatically and correctly answers the following questions with 1 for “yes” and 0 for “no”.

- a) Is c a capital letter ?
- b) Is m contained in $[1, 20]$ and is c character number m of w in this case ?
- c) Is m divisible by n or is n divisible by m ?
- d) Does $m = n$ and $n^2 < 17$ hold true ?
- e) Is one of the inequalities $m - n > m + n$ and $m \cdot n \geq -m \cdot n$ wrong ?
- f) Does the floating-point evaluation of m/n have first decimal digit 2 ?

Determine the corresponding correct answers $A \in \{0, 1\}$ solely by using arithmetic and logical operations stated in chapter 3 of the script and display them on the screen.² An exemplary input/output looks as follows:

```
Enter a character: c = u
Enter a word (<= 20 characters): w = house
Enter an integer: m = 3
and another one: n = -9
```

```
Is 'u' a capital letter ?
Answer: 0
```

```
Is 3 contained in [1,20] and is 'u' character number 3 of 'house' in this case ?
Answer: 1
```

```
Is 3 divisible by -9 or is -9 divisible by 3 ?
Answer: 1
```

```
Does 3 = -9 and -9*-9 < 17 hold true ?
Answer: 0
```

```
Is one of the inequalities 3 - -9 > 3 + -9 and 3*-9 >= -3*-9 wrong ?
Answer: 1
```

```
Does the floating-point evaluation of 3/-9 = -0.33... have first decimal digit 2 ?
Answer: 0
```

Find values m and n such that e) is answered with “no” and f) with “yes”.

²For f) you may also use explicit typecasting but you don't have to.

2. In the following program there are several arithmetical operations missing or incomplete (see script 3.1). Add them correctly at the positions marked by underscores _ so that the subsequent output will be produced. Explain your completion in detail!.

```
1  #include <stdio.h>
2  int main(void){
3      int x=27;
4      float y=-1.5;
5      char c1='U', c2;
6      char str[]="String",*s1=str,*s2;
7      y__; printf("\n y = %f",y);
8      x _= 7; printf("\n x = %d",x);
9      x *= 6; printf("\n %d",x _ 13);
10     c2 = c1__; printf("\n c1 = %c, c2 = %c",c1,c2);
11     s2 = s1__; printf("\n s1 = %s, s2 = %s ",s1,s2);
12     *s1 = *s2__; printf("\n *s1 = %c, *s2 = %c \n",*s1,*s2);
13     return 0;
14 }
```

Output:

```
y = -0.500000
x = 6
2
c1 = T, c2 = U
s1 = tring, s2 = String
*s1 = S, *s2 = S
```