

- c) Lesen Sie den berechneten Quotienten nach dem Rücksprung ab und kontrollieren Sie den Wert mit dem Taschenrechner: 1P

Berechneter Quotient	Ergebnis Taschenrechner

- d) Beschreiben Sie den Algorithmus und versuchen Sie zu zeigen, dass es sich bei diesem Algorithmus aus bedingter Subtraktion und Verschiebung tatsächlich um eine Division handelt¹. 2P

Aufgabe 2:

Gesucht ist ein korrekt arbeitendes C-Programm zur Berechnung von

$$I_D = K \cdot (U_{GS})^2$$

($K = 0,375$ – Konstante) mit Festkommaformaten für die Variablen:

	Bitanzahl	Nachkommabits	Wertebereich
I_D, U_{GS}, U_{GS}^2	16	8	-0x80 bis 0x7F,FF
K	16	15	-1 bis 0x0,FFFE

Das nachfolgende Beispiel enthält zusätzlich die korrekte Berechnung mit Gleitkommazahlen, aber bei den Festkommaberechnung wurde die Kommaverschiebungen gegenüber ganzzahligen Multiplikationen vergessen.

```

11 float float_K=0.375;
12 float float_ID, float_UGS=2.5;
13 int16_t fix_K, fix_ID, fix_UGS;
14
15 void main(){
16     fix_K=(int16_t)(float_K*32788);
17     fix_UGS=(int16_t)(float_UGS*255);
18     fix_UGS *= fix_UGS;
19     float_UGS *= float_UGS;
20     fix_ID = fix_K * fix_UGS;
21     float_ID = float_K * float_UGS;
22 }

```

- a) Bereiten Sie folgende Testbeispiele vor: 6P

Wert			Darstellung hex.		
U_{GS}	U_{GS}^2	I_D	U_{GS}	U_{GS}^2	I_D
1			0x0100		
2,5					
7,83					

¹Heben Sie sich diese Teilaufgabe bis zum Schluss auf. Die ist anspruchsvoll! Sie müssen darauf achten, welche Operation vor den Rotationen jeweils das Carry-Bit setzt oder löscht und was das Carry-Bit dann repräsentiert.

- b) Für welchen Wertebereich von U_{GS} funktioniert das Programm ohne Wertebereichsüberlauf?
1P
- c) Korrigieren Sie die Berechnungen der Festkommawerte in den Zeilen 18 und 20 so, dass richtige Ergebnisse herauskommen. 3P

Hinweis: Sie müssen jeweils 32-Bit-Produkte bilden und daraus die richtigen 16 Bit auswählen. Das erfordert eine zusätzliche Variable vom Typ `int32_t` und die Konvertierung eines Faktors vor jeder der beiden Multiplikation.