

Übungsaufgaben Blatt 6 - Operatoren und Ausdrücke

PG 1 - Einführung in die Programmierung mit C - Prof. Dr. Ruben Jubeh

Aufgabe 1 : Inkrementierung

Welche Werte haben die Integer Variablen nach Ausführung dieser Anweisungen? Warum?

```
1  int i = 10;  
2  int n = i++%5;
```

1. Welche Werte hätten Sie, wenn statt der Postfix-Inkrementierung eine Präfix-Inkrementierung ausgeführt worden wäre? Macht es einen Unterschied?
2. Was können Sie tun, um solche Situationen zu vermeiden?

Aufgabe 2 : Ans Telefon gehen oder nicht: answerPhone

Ihr Smartphone klingelt. Vervollständigen Sie die Funktion **answerPhone**, die ermittelt, ob Sie abnehmen sollen oder nicht: Die Funktion gibt **1** (d.h. **true**) zurück, wenn Sie abnehmen sollen und **0** (d.h. **false**), wenn Sie nicht abnehmen sollen. Im Normalfall nehmen Sie ab, außer morgens, dann nehmen Sie nur ab, wenn Ihre Mutter anruft. In allen Fällen, in denen Sie schlafen, nehmen Sie nicht ab.

```
1  /*  
2     Assumes all inputs are either 0 (false) or 1 (true).  
3     Returns 1 if we answer the phone.  
4  */  
5  int answerPhone(int isMorning, int isMom, int isSleeping) {  
6      //Your Code here  
7      return 1;  
8  }
```

Aufgabe 3 : SquirrelParty

Wenn Eichhörnchen eine Party feiern, rauchen sie gerne Zigarren. Eine Party ist ein Erfolg, wenn zwischen 40 und 60 Zigarren geraucht werden (inklusive der Grenzen).

Wird eine erfolgreiche Party am Wochenende gefeiert, dann gibt es für die Anzahl der gerauchten Zigarren keine Obergrenze. D.h. die Party ist immer ein Erfolg, wenn 40 Zigarren oder mehr geraucht wurden. Die Funktion erwartet die Anzahl der Zigarren (**numCigars**), sowie die Information, ob es Wochenende ist oder nicht in der Variable **isWeekend**: kein Wochenende = 0, Wochenende = 1.

Die Funktion gibt **1** zurück, wenn die Party ein Erfolg war, sie gibt **0** zurück, wenn die Party kein Erfolg war.

```

1  int squirrelParty(int numCigars, int isWeekend) {
2      //Your code here
3      return 1;
4  }

```

Aufgabe 4 : noThirteenSum

Schreiben Sie eine Funktion, die 3 Parameter als Integer erwartet. Geben Sie die Summe der drei Variablen zurück, außer, wenn einer der drei Werte 13 ist. In diesem Fall wird der Wert nicht in die Summe miteinbezogen und Werte *rechts* von dieser Variable zählen ebenfalls nicht mit zur Summe. Wenn **b** beispielsweise gleich 13 ist, dann gehen weder **b** noch **c** in die Berechnung der Summe ein.

```

1  int noThirteenSum(int a, int b, int c) {
2      //Your code here
3      return 1;
4  }

```

Aufgabe 5 : Komische Zählereien: sumNoTeens

Geben Sie die Summe von 3 Ganzzahlen zurück, außer einer der Werte repräsentiert ein Teenageralter zwischen 13 und 19 (inklusive). In diesem Fall zählt der Wert als 0. Es gilt die folgende Ausnahme: 15 und 16 zählen nichts als Teens und werden daher mit in die Summe eingerechnet.

Schreiben Sie eine Hilfsfunktion `int fixTeenAge(int n)`, die einen `int` Wert entgegennimmt und einen korrigierten Wert zurückgibt, den sie dann zur Summenberechnung verwenden können. Dadurch verhindern Sie, dass Sie diesen Code dreimal tippen müssen (*Don't Repeat Yourself (DRY) - Prinzip*).

```

1  int sumNoTeens(int a, int b, int c) {
2      //Your code here
3      return 1;
4  }

```

Aufgabe 6 : Du kommst hier nicht rein. Oder Vielleicht doch?

Zusammen mit Ihrer Verabredung wollen Sie in einen angesagten Club. Der Parameter **you** beschreibt die *Stylishness* Ihres Outfits zwischen 0 und 10 inklusive. Der Parameter **date** beschreibt die *Stylishness* Ihrer Verabredung im selben Wertebereich.

Das Ergebnis, ob Sie in den Club dürfen wird wie folgt kodiert: 0=nein, 1=vielleicht, 2=ja. Wenn einer von Ihnen sehr *stylish* ist (8 oder mehr), dann ist das Ergebnis 2, außer einer von Ihnen hat einen *style* von 2 oder weniger, dann ist das Ergebnis 0 (nein). In allen anderen Fällen ist das Ergebnis 1 (vielleicht).

```

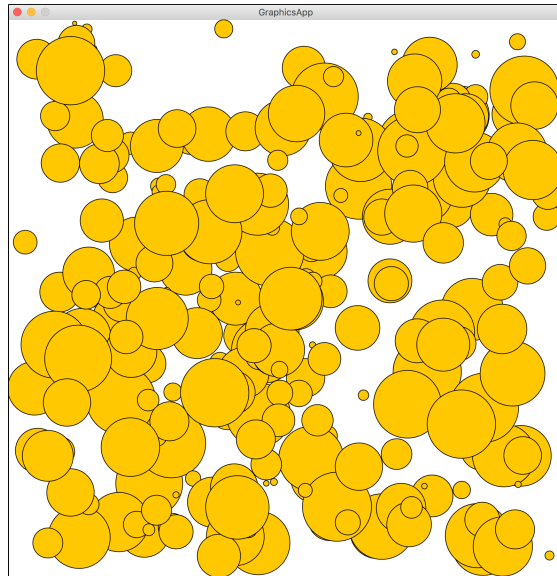
1  int checkAdmission(int you, int date) {
2      //Your code here
3      return 1;

```

Aufgabe 7 : Zufällige Kreise und Tastaturevents

Schreiben Sie eine *GraphicsApp*, die 250 Kreise mit unterschiedlichen Positionen und Durchmessern zeichnet. Dabei soll jeder Kreis einen zufälligen Durchmesser zwischen 5 und 100 Pixeln haben und eine zufällige Position innerhalb der Zeichenfläche. Achten Sie darauf, dass sich dabei der gesamte Kreis immer innerhalb der Zeichenfläche befindet.

Die folgende Abbildung zeigt einen Beispielaufruf des Programms:



Erweitern Sie Ihr Programm so, dass bei jedem Druck auf die Taste **n** (für neu), weitere Zufallskreise gezeichnet werden.

Verwenden Sie zur Berechnung der zufälligen Werte die Library *random.h*.

Schlagen Sie die Verwendung in der Dokumentation nach: <http://fbim.fh-regensburg.de/~hem38149/doc/cslib/>.