

Klausur Programmieren 2

HAW-Hamburg, Fakultät Technik und Informatik, Department Informations- und Elektrotechnik

Prof. Dr. Robert Heß, 3.2.2015, Bearbeitungsdauer: 180 Min.

Erlaubte Hilfsmittel: Vorlesungsunterlagen, Lösungen aus dem Praktikum und C/C++ Einführungsbücher.

Ergebnis: von 100 Punkten

Note: Punkte.

1 Einleitung

Für viele Anwendungen in der Computertechnik werden Zufallszahlen benötigt. Häufig werden dabei nicht *echte* Zufallszahlen verwendet, sondern nur Zahlenfolgen mit ähnlichen Eigenschaften wie zufällige Zahlen. Diese Zahlenfolge wird nicht als Ganzes zur Verfügung gestellt, sondern durch einen geeigneten Algorithmus berechnet.

Der *lineare Kongruenzgenerator* ermittelt für eine solche Zahlenfolge $(y_n) = (y_0, y_1, y_2, \dots)$ eine Zahl y_k aus der jeweils vorigen Zahl y_{k-1} mit folgender Gleichung:

$$y_k = (a y_{k-1} + b) \bmod m, \quad m \in \mathbb{N}_{>1}, \quad a, b \in \{1, 2, \dots, m-1\}$$

Werden die Konstanten a , b und m geeignet gewählt, so erfüllt die Zahlenfolge (y_n) die Anforderungen für die meisten Anwendungen.

Die Konstanten $a = 1\,664\,525$, $b = 1\,013\,904\,223$ und $m = 2^{32} = 4\,294\,967\,296$ erzeugen zum Beispiel eine Zahlenfolge, die sich erst nach 2^{32} Zahlen wiederholt und die meisten Anforderungen an Zufallszahlen erfüllt. Ein Vorteil dieser Konstanten ist, dass der Modulo-Divisor m genau dem Zahlenumfang einer 4 Byte **unsigned** Variablen entspricht. D.h. die Modulo-Division entfällt und die Gleichung reduziert sich zu:

$$y_k = (1\,664\,525 \cdot y_{k-1} + 1\,013\,904\,223)$$

Damit sich nach jedem Programmstart nicht immer exakt die selbe Zahlenfolge ergibt, wird üblicherweise der Anfangswert y_0 durch eine Funktion gesetzt, so dass sich alle folgenden Zahlen entsprechend anders ergeben. Dieser Anfangswert kann z.B. mittels aktueller Zeit gesetzt werden.

2 Programmieraufgaben

Aufgabe 1 (5 Punkte)

Legen Sie zunächst ein leeres Projekt an, bei dem die Funktionen zur Bestimmung der Zufallszahlen in eine separate Quellcode- und Header-Datei ausgelagert werden (z.B. *Rand.c* und *Rand.h*).

Aufgabe 2 (15 Punkte)

Implementieren Sie die Funktion **unsigned** Rand(), welche die nächste Zahl in der Zahlenfolge berechnet und zurückgibt. Verwenden Sie in der Quellcode-Datei für die Zufallszahlen (ausnahmsweise) eine globale Variable, damit der jeweils vorige Wert für die Berechnung zur Verfügung steht, und initialisieren Sie diese mit dem Wert null.

Aufgabe 3 (5 Punkte)

Implementieren Sie die Funktion `void sRand(unsigned Seed)`, mit der der Anwender gezielt die letzte Zufallszahl y_{k-1} setzen kann.

Aufgabe 4 (10 Punkte)

Implementieren Sie die Funktion `double RandU()`, welche eine Zufallszahl $0 \leq u_k < 1$ ermittelt und zurückgibt. Hinweis: $u_k = y_k/m$

Aufgabe 5 (20 Punkte)

Fragen Sie im Hauptprogramm einen Startwert sicher vom Benutzer ab, und initialisieren Sie damit den Zufallszahlengenerator (`sRand()`). Ermitteln Sie danach mit `RandU()` 1 000 Zufallszahlen u_k , und speichern Sie diese sequentiell ohne weitere Zusätze in eine Binärdatei mit Namen *Zufall.dat*.

Aufgabe 6 (15 Punkte)

Ermitteln Sie für $n = 1000$ Zufallszahlen u_k den Mittelwert μ_u , die Varianz σ_u^2 und die Standardabweichung σ_u gemäß folgender Formeln:

$$\mu_u = \frac{1}{n} \sum_{k=1}^n u_k \quad \sigma_u^2 = \frac{1}{n-1} \left\{ \sum_{k=1}^n u_k^2 - \frac{1}{n} \left(\sum_{k=1}^n u_k \right)^2 \right\} \quad \sigma_u = \sqrt{\sigma_u^2}$$

Es werden folgende Werte erwartet (auf fünf Stellen nach dem Komma gerundet):

$$\mu_u = 0,50000 \quad \sigma_u^2 = 0,08333 \quad \sigma_u = 0,28868$$

Aufgabe 7 (10 Punkte)

Testen Sie Ihr Programm, indem Sie zunächst die Zahlenfolge mit `sRand(x)` initialisieren und dann für 1000 Werte u_k den Mittelwert μ_u , die Varianz σ_u^2 und die Standardabweichung σ_u bestimmen. Geben Sie Ihre Ergebnisse für die Startwerte 0, 1 und 2 auf fünf Stellen nach dem Komma an.

sRand(x)	μ_u	σ_u^2	σ_u
0			
1			
2			

3 Verständnisfragen

Aufgabe 8 (8 Punkte)

a) Definieren Sie einen Zeigervektor für `int`-Variablen mit sechs Elementen.

.....

b) Definieren Sie einen Vektorzeiger für `int`-Vektoren mit sechs Elementen.

.....

c) Bestimmen Sie die Größe im Speicher auf unseren Systemen in Byte.

Variable a):

Variable b):

Aufgabe 9 (4 Punkte)

Was bezeichnet der Begriff *memory leakage*?

Aufgabe 10 (5 Punkte)

Definieren Sie einen Struktur für die Glieder einer doppelt verkettete Liste in die jeweils drei `double`-Variablen x , y und z gespeichert werden können.

Aufgabe 11 (3 Punkte)

Was passiert bei einer rekursiven Funktion ohne Abbruchbedingung?