Klausur zum Modul

Programmierung mit C++ 1

Studiengang	**Bachelor Informatik**
Klausurdatum	**09.02.2015 / 13.00 - 14.45 Uhr**
Schreibzeug	dokumentenechter Stift (kein Bleistift!)
zugelassene Hilfsmittel	*C++*–Bücher, Skript zur Vorlesung
nicht zugelassene Hilfsmittel	Rechner, Mobiltelefone, Aufgabensammlungen, alte Klausuren, Praktikumsblätter und deren Lösungen

Bewertung der Aufgaben:

Aufgabe	1	2	3	Σ
Punkte (max.)	10	10	10	30

---

1 Verwaltung von Speicherblöcken

Liste 1: Festlegungen zu Aufgabe 1

```
1  int block[] = { 0, 0, 15, 2, 0, 6, 7, 8, 13, 0, 0, 0, 0, 14, 3, -1, 0, 0, 0, 0 };
2
3  struct chain {
4    string       name;                // file name
5    unsigned int firstblock;          // number of the first block in block[]
6    unsigned int get_length();        // get length of chain
7    unsigned int get_last();          // get number of the last block in the chain
8  };       // ----------  end of struct chain  ----------
```

In dem Feld `block` (Liste 1) werden Blocknummern verwaltet. Die Blocknummern bilden eine Kette, die durch eine Struktur von Typ `chain` beschrieben wird. Eine solche Struktur enthält nur die Nummer des ersten Blockes einer Kette in der Variablen `firstblock`. Diese Nummer ist ein Index in das globale Feld `block`. Unter dem ersten Index in `block` steht die Nummer des zweiten Blocks und so weiter. Die Kette endet, wenn der Eintrag in `block` den Wert -1 hat. Abbildung 1 veranschaulicht die Kette, die in Liste 1 als Beispiel verwendet wird. Der erste Index ist in diesem Beispiel 5, die Nummer des 2. Blocks demnach 6 und so weiter.

- Schreiben Sie eine Memberfunktion `get_length`, die die Länge der Kette (Anzahl der zugehörigen Einträge in `block`) feststellt und zurückgibt. Dazu muss die gesamte Kette durchgelaufen werden. Der Block mit der Endemarkierung -1 ist immer Bestandteil der Kette und zählt also mit.
- Schreiben Sie eine Memberfunktion `get_last`, die den Index des letzten Blockes der Kette zurückgibt (im Beispiel: 15). Dazu muss ebenfalls die gesamte Kette durchgelaufen werden.

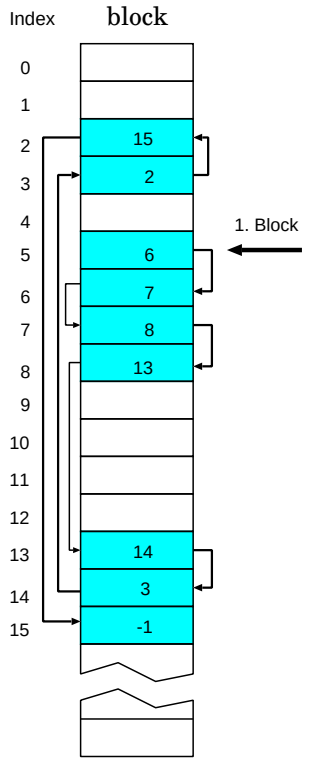

Abbildung 1: Verkettung in Liste 1, Zeile 2

2 Rekursion

Liste 2 zeigt die Definition und den Aufruf der rekursiven Funktion `rekursiv`.

1. Welchen Wert nimmt die Variable `erg` nach dem Funktionsaufruf im Hauptprogramm an ?
2. Welche Berechnung wird mit den Feldelementen von `a` durchgeführt ? Erläutern Sie die Berechnung anhand der 5 Feldwerte, indem Sie die Rekursion in Zeile 6 der Liste 2 auflösen:

$$a[4] + \frac{1}{2}(\ \ldots\ )$$

Liste 2: Rekursive Funktion `rekursiv`

```
1  double rekursiv ( double  *a, int n )
2  {
3    if ( n == 0 )
4      return a[0];
5    else
6      return a[n] + 0.5*rekursiv(a, n-1);
7  }   // -----  end of function rekursiv  -----
8
9  int main ( )
10 {
11   double  a[] = { 16, 8, 4, 2, 1 };
12   int     n   = sizeof(a)/sizeof(double);
13   double  erg = rekursiv( a, n-1 );            // Funktionsaufruf
14
15   cout << "Ergebnis = " << erg;
16   return 0;
17 }   // ----------  end of function main  ----------
```

3 ISBN-Nummer prüfen

Zur Überprüfung einer 10-stelligen ISBN-Nummer sind die ersten 9 Ziffern der Reihe nach von links mit den Faktoren $1, 2, 3, \ldots, 9$ zu multiplizieren und aufzusummieren. Das Ergebnis wir durch 11 geteilt. Der Divisionsrest muss der 10. Ziffer entsprechen. Wenn diese 10. Ziffer den Wert 10 besitzt, wird sie durch das Zeichen `'x'` dargestellt, sonst durch die Ziffern 0...9 . Für die korrekte ISBN-Nummer *349913599***x** ergibt sich dann folgende Berechnung:

$$1*3+2*4+3*9+4*9+5*1+6*3+7*5+8*9+9*9 \ = \ 285 \qquad (1)$$

$$285 \mod 11 \ = \ 10 \ (=\texttt{'x'}) \qquad (2)$$

Schreiben Sie eine Funktion `ISBN10_check` (Prototyp in Liste 3), die eine ISBN-Nummer übernimmt, aus den ersten 9 Ziffern die Prüfziffer berechnet und das Ergebnis mit der im Feld `number` übergebenen Prüfziffer vergleicht. Wenn die ISBN-Nummer richtig ist wird `true` zurückgegeben, sonst `false`. Beachten Sie, dass die ISBN-Nummer in einem `char`-Felder übergeben wird.

Liste 3: Prototyp der Funktion `ISBN10_check` und Aufruf in einem Hauptprogramm

```
1  bool ISBN10_check ( char *number );            // Prototyp
2
3  int main ( )
4  {
5    char  isbn1[] = { '3', '4', '9', '9', '1', '3', '5', '9', '9', 'x' };   // Beispiel
6    cout << " : " << boolalpha << ISBN10_check( isbn1 ) << "\n";
7    return 0;
8  }   // ----------  end of function main  ----------
```