

Es gab insgesamt 90 Punkte und 90 Minuten Zeit.

C- Teil

1. Welche der beiden C-Deklarationen bzw. Anweisungen führt beim Kompilieren zu einem Typfehler? Welche Anweisung führt nicht zu einem Typfehler? Begründen sie ihre Antwort. (3 Punkte)

1. `char c = 'a', *cptr; *cptr = c;`
2. `unsigned long u = 42, *uptr = u;`

2. Seien `a` und `b` Zeiger auf aufsteigend sortierte `int`-Arrays der Längen `alen` und `blen`. Implementieren Sie eine C-Funktion

```
int *merge_arrays(const int *a, unsigned long alen,
                  const int *b, unsigned long blen)
```

die einen Zeiger auf ein neues Array der Länge `alen+blen` liefert, in dem die Werte aus den Arrays, auf die `a` und `b` zeigen, aufsteigend sortiert sind. Die Funktion soll eine Laufzeit proportional zu `alen+blen` haben. Beachten Sie, dass `alen` und `blen` ungleich sein können und behandeln Sie diesen Fall entsprechend.

(8 Punkte)

3. Implementieren Sie, basierend auf der Funktion `qsort` aus der C-Standardbibliothek, eine C-Funktion

```
void sort_string_int_pairs(StringIntPair *tab, unsigned long len)
```

zum Sortieren eines Arrays über dem folgenden Basistyp:

```
typedef struct
{
    char *strptr;
    int intval;
} StringIntPair;
```

Dabei ist `len` die Länge des Arrays, das durch `tab` referenziert wird. Die `StringIntPair`-Strukturen sollen absteigend nach den `\0`-terminierten Strings sortiert werden, auf die jeweils die `strptr`-Komponente zeigt. Für den Vergleich zweier Strings sollen Sie die passende Funktion aus der C-Standard-Bibliothek verwenden. Falls die Strings für zwei zu vergleichenden `StringIntPair`-Strukturen gleich sind, soll aufsteigend nach der zweiten Komponente `intval` sortiert werden.

Zur Erinnerung geben wir hier nochmal den Typ der Funktion `qsort` an:

```
void qsort(void *base, size_t num_elems, size_t size_elem,
           int (*compare)(const void *, const void *));
```

(8 Punkte)

4. Die folgenden Funktionen prozessieren 3-dimensionale Arrays mit dem Basistyp `double`. Solche Arrays sind z.B. geeignet, Eigenschaften von Punkten im 3-dimensionalen Raum zu repräsentieren.

```
void add_sum_3D(double ***c, double ***a, double ***b, int m, int n, int l) {
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            for (int k = 0; k < l; k++)
                c[i][j][k] += (a[i][j][k] + b[i][j][k]);
}
```

```
void add_product_3D(double ***c, double ***a, double ***b, int m, int n, int l) {
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            for (int k = 0; k < l; k++)
                c[i][j][k] += (a[i][j][k] * b[i][j][k]);
}
```

Offensichtlich besteht der einzige Unterschied der beiden Funktionen in der Operation, die in der innersten Schleife auf die Werte aus den Arrays `c`, `a` und `b` angewendet wird. In dieser Aufgabe geht es darum, von dieser Operation zu abstrahieren, indem man einen Funktionszeiger `f` von einem geeigneten Typ `CombinerFunc` als Parameter verwendet und den Zeiger je nach Operation passend instantiiert. Geben Sie eine geeignete Typdeklaration für `CombinerFunction` an und implementieren Sie eine Funktion

```
void add_generic_3D(CombinerFunction f, double ***c,
                  double ***a, double ***b, int m, int n, int l)
```

die in der innersten Schleife die Funktion `f` auf die relevanten Wert anwendet. Es sollen Funktionen `add_sum` und `add_product` implementiert werden, so dass die Funktionsaufrufe

```
add_generic_3D(add_sum, c, a, b, m, n, l);
add_generic_3D(add_product, c, a, b, m, n, l);
```

das gleiche berechnen, wie die oben angegebenen Funktionen (wenn die gleichen Parameter außer `add_sum` bzw. `add_product` verwendet werden).

(12 Punkte)

5. Schreiben Sie eine C-Funktion

```
char *random_sequence_prob(const char *alphabet, const double *prob, size_t n)
```

die eine Zufallssequenz der Länge `n` über dem Alphabet `alphabet` in einer Laufzeit, die proportional zu `n * k` ist, berechnet. `k` ist die Größe des Alphabets, d.h. die Länge des `\0`-terminierten Strings `alphabet`. Dieser enthält keine Duplikate. `prob` ist ein Zeiger auf einen Speicherbereich mit genau `k` `double`-Werten, deren Summe 1 ist. Für das `i`-te Zeichen des Alphabets mit $0 \leq i \leq k - 1$ und für alle Positionen ist die Wahrscheinlichkeit des Vorkommens `prob[i]`.

Der Rückgabewert der Funktion soll ein Zeiger auf einen Speicherbereich mit der Zufallssequenz sein. Die Sequenz ist nicht `\0`-terminiert. Benutzen Sie die Funktion `drand48()` zum Generieren von Zufallszahlen. Sie können davon ausgehen, dass `drand48()` konstante Zeit benötigt. Um die Initialisierung des Seeds für den Zufallsgenerator brauchen Sie sich nicht zu kümmern.

Beispiel: Falls `alphabet` auf den String "acgt" zeigt und `prob` auf das Array mit den Werten 0.1, 0.2, 0.3, 0.4, dann liefert `random_sequence_prob(alphabet, prob, n)` eine Zufallssequenz der Länge `n` in der `a` etwa $0.1n$ mal vorkommt, `c` etwa $0.2n$ mal vorkommt, `g` etwa $0.3n$ mal vorkommt, und `t` etwa $0.4n$ mal vorkommt.

(10 Punkte)

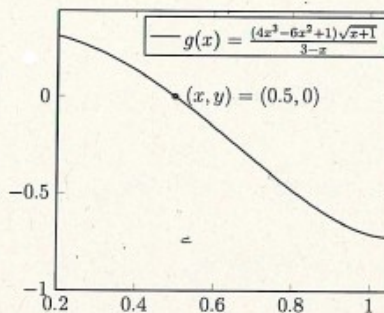
6. Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ eine Funktion und sei $\ell, r \in \mathbb{R}$ mit $\ell \leq r$, so dass $f(x) \geq f(x')$ für alle $\ell \leq x \leq x' \leq r$. D.h. f ist monoton fallend im Intervall $[\ell, r]$. Es geht nun darum, für ein gegebenes $y \in \mathbb{R}$ mit $f(r) \leq y \leq f(\ell)$ ein $x \in \mathbb{R}$ zu bestimmen, so dass $f(x) = y$ ist. Durch die Monotonieeigenschaft kann man das gesuchte x effizient mit Hilfe einer binären Suche berechnen.

Schreiben Sie eine C-Funktion

```
double invert_decreasing_function(double (*f)(double), double l,
                                double r, double y)
```

die für eine Funktion f , die monoton fallend ist im Intervall $[l, r]$ und ein y einen Wert x mit einer return-Anweisung zurückliefert, so dass $f(x) = y$. Verwenden Sie eine binäre Suche, die in jedem Schritt ein neues Teilintervall, in dem der gesuchte Wert liegt, berechnet. Die Iteration soll stoppen, wenn die Breite des Teilintervalls kleiner als eine vorgegebene Konstante TOL ist. In diesem Fall soll die Intervallmitte zurückgegeben werden.

Beispiel: $g(x) = \frac{(4x^3 - 6x^2 + 1)\sqrt{x+1}}{3-x}$ ist monoton fallend im Intervall $[0.2, 1.0584]$. Für $y = 0$ ist $g(0.5) = y$. Die zu implementierende Funktion muss also beim Aufruf mit den Parametern $g, 0.2, 1.0584, 0$ den Wert 0.5 zurückliefern.



(12 Punkte)

7. Wenn man beim n-Queens Problem nicht nur die Anzahl der Lösungen zählen will, sondern auch noch die Lösungen selbst aufzählen möchte, dann muss man jede Platzierung einer Königin an Position (i, j) protokollieren. Wie kann man diese Positionen auf einfache Weise protokollieren, ohne Paare von Positionen zu speichern? Wie kann man dabei Zeilen erkennen, in denen noch keine Königin platziert wurde? (2 Punkte)

C++ Teil

8. Der Absolutwert einer komplexen Zahl $x = a + bi$ ist $|x| = \sqrt{a^2 + b^2}$. Für zwei komplexe Zahlen $x = a + bi$ und $y = c + di$ ist das Produkt xy definiert durch $xy = ac - bd + (bc + ad)i$. Seien komplexe Zahlen durch die Klasse `Complex` in C++ implementiert:

```
class Complex
{
private:
    int real, imag;
public:
    Complex(int r,int i) {
        real = r;
        imag = i;
    }
    std::string show() const {
        return std::string("(") + std::to_string(real) + std::string("+") +
            std::to_string(imag) + std::string("i");
    }
};
```

Dabei liefert die Methode `std::to_string` für eine Zahl und ein String-Literal die String-Darstellung in Form einer Instanz der Klasse `std::string`. Implementieren Sie innerhalb dieser Klasse zwei inline-Methoden

`double absolute(void) const`

`Complex operator* (const Complex &other) const`

zur Berechnung des Absolutwertes einer komplexen Zahl und zur Multiplikation zweier komplexer Zahlen. Da diese Methoden innerhalb der Klasse implementiert werden, können Sie auf die privaten Member-Variablen zugreifen. Die Methoden sollen so implementiert werden, dass das folgende Programm die daneben stehende Ausgabe liefert:

```
Complex a = Complex(2,4);
std::cout << a.absolute() << std::endl;
Complex b = Complex(3,1);
std::cout << a.show() << " * " << b.show() << " = "
<< (a * b).show() << std::endl;
```

(5 Punkte)

```
4.47214
(2+4i) * (3+1i) = (2+14i)
```

9. Wie zählt man typischerweise die Elemente eines Containers aus der C++- Standard-Template Library auf? Geben Sie entsprechenden C++-Programmcode an, der das Vorgehen beispielhaft zeigt. (3 Punkte)

Aufgaben zu Shared Memory Multithreading

10. Die rekursive Funktion

```
size_t f(size_t n) {
    return n <= 1 ? 1 : (f(n-1) + f(n-2));
}
```

soll auf alle Werte in einer Tabelle `table` von `num_tasks` ganzen Zahlen angewendet werden, und zwar so, dass an der *i*-ten Stelle in der Tabelle nach Ausführung von `f` der Wert `f(i)` steht.

Wenn man nur einen Thread verwendet, kann man z.B. die folgende Schleife implementieren:

tieren:

```
for (size_t i = 0; i < num_tasks; i++)  
table[i] = f(i);
```

Hier geht es aber darum, die Werte in `table` mit mehreren Threads gleichzeitig zu berechnen. In der Vorlesung wurde gezeigt, wie man eine Funktion

```
void pfn_run_threaded(size_t k, size_t num_tasks,  
PfnThreadFunc thread_proc, void *thread_data);
```

implementiert, die eine Funktion `thread_proc` mit `k` Threads auf Werte zwischen 0 und `num_tasks-1` anwendet. Jeder dieser Werte kann als Nummer einer Aufgabe interpretiert werden. Der Typ `PfnThreadFunc` ist dabei wie folgt definiert:

```
typedef void (*PfnThreadFunc)(size_t thread_id, size_t task_num,  
void *thread_data);
```

Es wird nun jede Aufgabennummer als Index in `table` interpretiert, an dem zunächst das Argument `n` für die Funktion `f` steht und dann der Funktionswert `f(n)` gespeichert wird.

Implementieren Sie nun eine Funktion `eval_f`, so dass

```
pfn_run_threaded(k, num_tasks, eval_f, (void *) table);
```

die Funktion `f` mit `k` Threads auf `num_tasks` Werte in `table` anwendet. Es soll das gleiche Ergebnis entstehen wie bei der Berechnung mit einem Thread.

(9 Punkte)

Aufgaben zur r-Programmierung

11. Welchen Datentyp hat die Variable `a` nach der folgenden Wertzuweisung in R? (1 Punkt)
`a = c(1.0, 2.0, 3.0)`

12. Sei `a` ein Vektor mit numerischen Werten. Das folgende R-Skript berechnet die Summe der Elemente in `a`.

```
s ← 0  
for (x in a) {  
  s ← s + x  
}
```

Schreiben sie eine einzelne Anweisung in R, die das gleiche Ergebnis berechnet. (1 Punkt)

13. Gegeben sei die folgende R-Anweisung:

```
x ← c(1, 2, 3, 4, 5)
```

Geben sie einen syntaktisch korrekten Ausdruck in R an, der einen Vektor mit allen Elementen aus `x`, die größer als 3 sind, liefert. (1 Punkt)

14. Gegeben sei die folgende R-Anweisung:

```
df ← read.table("data.txt", header = TRUE)
```

Welchen Datentyp hat `df`? (1 Punkt)

15. Evaluieren sie den folgenden Ausdruck algebraisch, nicht numerisch:

$$\lim_{x \rightarrow 0} \ln(x+1)$$

16. Wir betrachten das folgende Fragment eines R-Skriptes:

```

f ← function (x) {
  z ← 0
  for (y in x) {
    z ← z + y
  }
  z/length(x)
}

k ← f(c(1, 2, 3, 4, 5))

```

Welchen Wert hat k ? (1 Punkt)

17. Die Funktion f aus der vorherigen Aufgabe kann man durch eine vordefinierte Funktion aus R ersetzen. Benutzen sie die passende vordefinierte Funktion, um k zu berechnen. (2 Punkte)

18.

Sei a ein Vektor mit n numerischen Werten. Schreiben sie einen mathematischen Ausdruck auf, der nach Auswertung, den Wert der Variable m liefert, der durch die folgende R-Anweisung berechnet wird. Sie dürfen in ihrem Ausdruck nicht die Logarithmus-Funktion verwenden. (2 Punkte)

$m = \exp(\text{sum}(\log(a)))$

19. Wir betrachten die folgende Anweisung in R:

$a \leftarrow \text{replicate}(100, f(m,n))$

Welche syntaktischen Eigenschaften müssen die Bezeichner f, m und n haben? (1 Punkt)

Was macht *replicate*? (1 Punkt)

20. Eine Krebsdiagnose hat eine falsch-positiv Rate von 5% und eine falsch-negativ Rate von 0.1%. Der Krebs hat eine Prävalenz von 1 zu 1000 in der Bevölkerung. Wie hoch ist die Rate positiver Tests in der gesamten Bevölkerung? (2 Punkte)

Wenn bei ihnen Krebs diagnostiziert wurde, wie hoch ist die Wahrscheinlichkeit, dass Sie Krebs haben? Geben Sie die Antwort in exakter Form, ohne einen Taschenrechner zu verwenden. (2 Punkte)

21. Ich möchte die Anzahl der Fische in einem See schätzen. Ich fange 100 Fische und markiere sie. Ich werfe sie zurück in den See und fange einen Tag später 100 Fische. Von ihnen sind 5 markiert. Ich schätze, es gibt 2 000 Fische im See. Ich benutze einen Bayes-Ansatz und die statistische Modellierung, um meine Fehler abzuschätzen.

Ein alter Fischer sagt, er habe die Fische kürzlich gezählt und 3 000 mit einer Standardabweichung von 100 gefunden. Wie würden Sie diese zusätzliche Information in einer Stichprobensimulation im Bayes-Kontext kombinieren? (1 Punkt)