

Gedächtnisprotokoll PfN2 Klausur – Ersttermin – 2022

1. (1) Identifizieren und korrigieren Sie den logischen Fehler in der folgenden C-Anweisung:

```
if (c < 7);  
    printf("c ist kleiner als 7\n");
```

2. (3) Beschreiben Sie kurz, was die folgenden Zeilen in einem Makefile bedeuten.

```
CC?=clang  
CFLAGS=-g -Wall -Werror -O3  
  
%.o: %.c  
    ${CC}  $<  -c  ${CFLAGS}  -o  $@
```

Nehmen wir an, man führt im aktuellen Verzeichnis den Befehl `ls -l` aus und es erscheint folgende Ausgabe:

```
-rw-r----- 1 joe staff 740 Jul 22 08:47 program.c  
-rw-r----- 1 joe staff 5140 Jul 22 08:52 program.o  
-rw-r----- 1 joe staff 9124 Jul 22 07:27 Makefile
```

Nehmen wir an, dass die Datei `Makefile` den oben gezeigten Inhalt hat. Was passiert, wenn man im Terminal den Befehl `make program.o` ausführt? Begründen Sie ihre Antwort.

3. (2) Gegeben seien die folgende Deklaration eines Funktionstyps `FuncType` und einer Funktion `func`, die einen Parameter vom Typ `FuncType` hat:

```
typedef int *(*FuncType) (const char *, char *,int,double);  
int func(FuncType f)
```

Geben Sie den Kopf einer Funktion `funcparam` an, so dass

```
return func(funcparam) == -1 ? EXIT_FAILURE : EXIT_SUCCESS;
```

eine korrekte C-Anweisung ist.

4. (7) Implementieren Sie, basierend auf der Funktion `qsort` aus der C-Standardbibliothek, eine C-Funktion

```
void sort_dates(Date *tab, size_t len)
```

zum Sortieren eines Arrays von Datumswerten, die jeweils aus einer Jahreszahl

und einer Monatsangabe bestehen. Letztere ist ein String aus dem folgenden Array:

```
static const char *months[] = {"Jan", "Feb", "Mar",  
                                "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
```

Die Struktur *Date* ist wie folgt deklariert.

```
typedef struct {  
    int year;  
    const char *month;  
} Date;
```

Das Array, das durch den Zeiger `tab` referenziert wird und `len` Einträge hat, soll aufsteigend nach den Jahreszahlen sortiert werden. Bei gleichen Jahreszahlen muss die Sortierung aufsteigend nach den Monaten im Jahr erfolgen, d.h. unter Berücksichtigung der Ordnung Jan<Feb...<Dec.

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 *));
```

5. (8) In der Vorlesung wurde das Divide and Conquer-Verfahren zur Multiplikation einer $m \times n$ -Matrix A mit einer $n \times l$ -Matrix B im Detail erläutert. Beschreiben Sie dieses Verfahren, gerne auch durch eine Skizze. Pseudocode ist nicht notwendig, aber die Teilungsschritte sollen dabei verständlich erläutert werden.

6. (8) Sei die Zahlenfolge $s_0, s_1, s_2, s_3, \dots$ definiert durch $s_0 = 1$ und $s_i = s_{i-1} * (2 * (2i - 1)) / (i + 1)$ für alle $i \geq 1$.

Implementieren Sie eine C-Funktion `enumerate_series()`, die diese Folge zeilenweise bis zum maximalen Folgeelement s_i , das sich durch eine Variable vom Typ `uint32_t` repräsentieren lässt, ausgibt. Der maximale Wert einer Variable vom Typ `uint32_t` ist `UINT32_MAX`. Verwenden Sie für den Wert, mit dem s_{i-1} multipliziert wird, eine Variable vom Typ `double`. Alle anderen Variablen sind vom Typ `uint32_t`.

7. (4) Schreiben Sie in C++ ein Funktionstemplate `swap` für das Vertauschen der Werte von zwei Variablen, deren Typ als Template-Parameter deklariert wird. Verwenden Sie Referenzen für die Parameter, aber keine Zeiger.
8. (16) In einer spärlich besetzten Matrix sind die meisten Werte 0 und nur wenige Werte verschieden von 0. Für eine spärlich besetzte $m \times n$ -Matrix ist die übliche explizite Repräsentation aller Werte durch ein Array mit $m \cdot n$ -Werten nicht sinnvoll. Daher werden für eine spärlich besetzte Matrix M nur die Werte

$M(r,c) \neq 0$ repräsentiert. Man benötigt daher eine Datenstruktur, um Indexpaare von Zeilen- und Spaltenindices mit einem Wert $\neq 0$ zu assoziieren. Implementieren Sie in C++ eine Klasse `SparseMatrix` mit einer privaten Membervariablen `non_zero_values`, einem Konstruktor `SparseMatrix(void)` und public Methoden `int get(size_t r, size_t c) const` und `void set(size_t r, size_t c, int value)` zum Lesen bzw. Schreiben von Matrixwerten vom Typ `int` für die angegebenen Zeilen- und Spaltenindices `r` und `c` und dem zu schreibenden Wert `value`. Die Methoden müssen so implementiert werden, dass das folgende Programm fehlerfrei läuft.

```
SparseMatrix m{};

const size_t rows = 5, columns = 5;
for (size_t r = 0; r < rows; r++)
    m.set(r,r,static_cast<int>(r+1));
for (size_t r = 0; r < rows; r++)
    for (size_t c = 0; c < columns; c++)
        if (r == c)
            assert(m.get(r,c) == static_cast<int>(r+1));
        else
            assert(m.get(r,c) == 0);
```

Verwenden Sie für `non_zero_values` einen passend STL-Container, um Indexpaare mit den Indexwerten $\neq 0$ zu assoziieren, so dass Sie keine eigenen Schleifen implementieren müssen.

9. (4) Wir betrachten die folgende Deklaration eines mit allen Schlüsselworten der Programmiersprache C initialisierten C++-Vektors.

```
const std::vector<std::string>
    ckeywords{"auto", "double", "int", --- hier sich jetzt 28 weitere
              Einträge denken --- "while"};
```

Schreiben Sie C++-Programmcode, der die Anzahl der Schlüsselworte, sowie die Summe der Längen aller Schlüsselworte berechnet und ausgibt. Für eine Instanz `s` der Klasse `std::string` liefert `s.size()` die Länge des Strings.

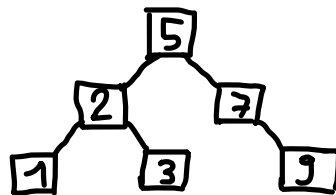
10. (7) In den Übungen haben Sie Techniken zur Implementierung binärer Bäume in C kennengelernt. Hier sehen Sie den Anfang der Deklaration einer C++-Klasse, die auf den gleichen Techniken basiert. Aus Platzgründen zeigen wir nur den Konstruktor, die private rekursive Methode `print_rec` und ihren Aufruf in der public Methoden `print` zur Ausgabe des Binärbaums.

```
template<typename T>
class Bintree
{
    struct Node {
        Node *left, *right;
        size_t count;
        T data;
        Node(const T &_data)
            : left(nullptr), right(nullptr), count(1), data(_data)
        {}
    };
    void print_rec(size_t d, const Node *node) const {
        if (node != nullptr) {
            print_rec(d+1, node->right);
            for (size_t i = 0; i < 3*d; i++) std::cout << " ";
            std::cout << node->data << " " << node->count <<
            std::endl;
        }
    };
};
```

```
Node *root;
public:
Bintree(void) : root(nullptr) {}
void print(void) const { print_rec(0, root); }
```

Implementieren Sie in dieser Klasse eine Methode `max_depth()`, die mit einer `return`-Anweisung die maximale Tiefe eines Knotens des durch `root` repräsentierten Binärbaums liefert. Die Tiefe der Wurzel ist 0. Die Kinder der Wurzel haben Tiefe 1 etc. Für den unten dargestellten Baum soll der Rückgabewert 2 sein. Da die Methode keinen Parameter hat, kann sie selbst nicht rekursiv sein, aber sie kann eine rekursive private Methode aufrufen, die Sie implementieren müssen. Sie können die Deklaration einer Funktion `std::max` zur Berechnung des Maximums zweier Werte des gleichen Typs voraussetzen.

Darstellung eines Binärbaums mit Werten vom Typ `T=int`. Die `count`-Werte sind nicht dargestellt.



11. (1) Definieren Sie den Begriff *Thread*, wie er in der Informatik verwendet wird.
12. (12) Das folgende Programm wurde entwickelt, um die Häufigkeiten der Vorkommen von allen möglichen Zeichen in einem String zu ermitteln.

```

void count_characters(size_t *counters, const char *sptr,
                    size_t len) {
    for (size_t idx = 0; idx < len; idx++)
        counters[static_cast<size_t>(sptr[idx])]++;
}

void count_characters_threaded(const char *str,
                             size_t strlength,
                             size_t num_threads) {
    size_t counters[256] = {0};
    std::vector<std::thread> threads{};
    const size_t width = strlength/num_threads;
    for (size_t t = 0; t < num_threads; t++)
        threads.push_back(std::thread(count_characters, counters,
                                      str + t * width,
                                      t < num_threads - 1
                                      ? width
                                      : (strlength - t * width)));
    for (auto &&th : threads) th.join();
    for (size_t idx = 0; idx < 256; idx++)
        if (counters[idx] > 0)
            std::cout << idx << "\t" << counters[idx] << std::endl;
}

```

Falls `num_threads > 1`, dann erfolgt eine Berechnung in mehreren Threads. Allerdings liefert das Programm dann mit hoher Wahrscheinlichkeit die falschen Zählwerte. Erläutern Sie, warum das so ist und beschreiben Sie, was man ändern muss, damit die Zählwerte korrekt berechnet werden. Bewerten Sie Ihre Änderung bzgl. ihrer Effizienz.

13. (2) In R ist `%%` der Modulo-Operator.

- Was ist der Datatype der folgenden Funktion `p()`?
- Was ist `p(8)`?
- Mit einfachen Worten: Was macht die folgende Funktion `p()`?

```
p <- function (n) {  
  if (n<=2) {  
    return (TRUE)  
  }  
  for (i in 2:sqrt(n)) {  
    if ((n %% i) == 0) {  
      return (FALSE)  
    }  
  }  
  return (TRUE)  
}
```

14. (1) Die folgende Funktion `f()` gibt einen Vektor zurück, in dem jedes Element das Quadrat des Elements von `v` ist.

```
f <- function(v) {  
  r <- vector()  
  for (x in v) {  
    r <- c(r, x * x)  
  }  
  return(r)  
}
```

Schreiben Sie eine schnellere Version von `f()`, die Vektoroperationen verwendet und nicht mehr als vier Zeilen lang ist.

15. (1) 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.

16. (1) Gegeben sei die folgende R-Anweisung:

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

Welchen Datentyp hat df?

17. (1) Was bedeutet das Symbol λ in einer exponentiellen Verteilung der Form $\lambda e^{-\lambda t}$?

18. (1) Nennen Sie eine von zwei Eigenschaften von Ereignissen, deren Verteilung einer Poisson-Verteilung entspricht.

19. (2) Evaluieren Sie folgenden Ausdruck algebraisch, nicht numerisch:

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

20. (1) 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?

21. (4) Aus einer nichtlinearen Regression auf $y = e^{-\beta x} \sin(2\pi\omega + \varphi)$ bekam ich einen Wert von $\varphi = 7$. Schreiben Sie eine R-Funktion, die ein beliebiges x nimmt und in einen Bereich von $(0 \dots 2\pi)$ verschiebt. R hat eine Variable pi mit einem vordefinierten Wert. Verwenden Sie diesen Rahmen

```
squash_to_range <- function (x) {  
  ..  
  return(x)  
}
```

22. (1) Wir betrachten den folgenden Ausdruck, in dem β eine positive Zahl ist:

$$y = e^{-\beta x} \sin(2\pi\omega + \varphi)$$

Was ist die Bedeutung bzw. der Einfluss von β (für $\beta > 0$) in diesem Ausdruck?

23. (2) Wir betrachten die folgende Anweisung in R:

```
a <- replicate(100, f(m, n))
```

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

Welchen Datentyp hat a ? (1 Punkt)