

Name, Vorname: .....

Legi-Nummer: .....

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte, und dass ich die allgemeinen Richtlinien gelesen und verstanden habe.

*I confirm with my signature that I was able to take this exam under regular conditions and that I have read and understood the general guidelines.*

Unterschrift:

**Allgemeine Richtlinien:**

**General guidelines:**

1. Dauer der Prüfung: 60 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für gesprochene Sprachen). Keine eigenen Notizblätter! Bei Bedarf stellen wir Ihnen weitere Blätter zur Verfügung.
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen sind deutlich durchzustreichen! Korrekturen bei Multiple-Choice Aufgaben bitte unmissverständlich anbringen! Lösungen auf Notizblättern werden nur in Notfällen berücksichtigt. Bitte kontaktieren Sie in diesem Fall eine Aufsichtsperson.
5. Es gibt keine Negativpunkte für falsche Antworten.
6. Störungen durch irgendjemanden oder irgendetwas melden Sie bitte sofort der Aufsichtsperson.
7. Wir sammeln die Prüfung zum Schluss ein. Wichtig: stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einem Assistenten eingezogen wird. Stecken Sie keine Prüfung ein und lassen Sie Ihre Prüfung nicht einfach am Platz liegen. Dasselbe gilt, wenn Sie früher abgeben wollen: bitte melden Sie sich lautlos, und wir holen die Prüfung ab. Vorzeitige Abgaben sind nur bis 15 Minuten vor Prüfungsende möglich.
8. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen.
9. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 60 minutes.*
- Permitted examination aids: dictionary (for spoken languages). No sheets of your own! We will give you extra sheets on demand.*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.*
- Solutions must be written directly onto the exam sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be crossed out clearly. Provide corrections to answers of multiple choice questions without any ambiguity! Solutions on extra sheets will be considered only in emergencies. In this case, please contact a supervisor.*
- There are no negative points for false answers.*
- If you feel disturbed by anyone or anything, let the supervisor of the exam know immediately.*
- We collect the exams at the end. Important: you must ensure that your exam has been collected by an assistant. Do not take any exam with you and do not leave your exam behind on your desk. The same applies when you want to finish early: please contact us silently and we will collect the exam. Handing in your exam ahead of time is only possible until 15 minutes before the exam ends.*
- If you need to go to the toilet, raise your hand and wait for a supervisor.*
- We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.*

Aufgabe	1	2	3	4	5	6	7	8	Σ
Punkte									
Maximum	9	8	6	8	8	8	7	8	62

Viel Erfolg!

*Good luck!*

## **Aufgaben / *Tasks***

<b>1</b>	<b>Typen und Werte (<i>9 Punkte</i>)</b>	<b>3</b>
<b>2</b>	<b>Schleifenausgaben (<i>8 Punkte</i>)</b>	<b>4</b>
<b>3</b>	<b>Primzahlzerlegung (<i>6 Punkte</i>)</b>	<b>5</b>
<b>4</b>	<b>Zeiger und Bereiche (<i>8 Punkte</i>)</b>	<b>6</b>
<b>5</b>	<b>Normalisierte Fließkommazahlen (<i>8 Punkte</i>)</b>	<b>7</b>
<b>6</b>	<b>Rekursion (<i>8 Punkte</i>)</b>	<b>8</b>
<b>7</b>	<b>EBNF (<i>7 Punkte</i>)</b>	<b>10</b>
<b>8</b>	<b>Sortierte Liste (<i>8 Punkte</i>)</b>	<b>12</b>

# 1 Typen und Werte (9 Punkte)

Geben Sie für jeden der sechs Ausdrücke unten jeweils C++-Typ (0.5 P) und Wert (1 P) an! Nehmen Sie für Fließkommazahlen den Standard IEEE 754 an! *For each of the six expressions below, provide the C++ type (0.5 P) and value (1 P)! For floating point numbers, assume the IEEE 754 standard!*

---

(a)  $5 / 2 * 2.0$

1.5 P

Typ/Type

Wert/Value

(b)  $0xa + 0xb$

1.5 P

Typ/Type

Wert/Value

(c)  $0 / 1 == 0 \ || \ 1 / 0 == 1$

1.5 P

Typ/Type

Wert/Value

(d)  $1.0e4f / 1.0e2f$

1.5 P

Typ/Type

Wert/Value

(e)  $3 + 17 \% 4$

1.5 P

Typ/Type

Wert/Value

(f)  $0.1f == 0.1$

1.5 P

Typ/Type

Wert/Value

## 2 Schleifenausgaben (8 Punkte)

Geben Sie für jedes der folgenden vier Code-Stücke die Ausgabe aus, die generiert wird! Nehmen Sie für Fließkommazahlen den Standard IEEE 754 an! Falls es sich um eine Endlosschleife handelt, schreiben Sie "Endlosschleife"! *For each of the following four code snippets, provide the output that is being generated! For floating point numbers, assume the IEEE 754 standard! In case of an infinite loop, write "infinite loop"!*

---

(a) 

```
int a[3] = {2, 3, 1};
for (int i = 0; i < 3; i = a[i])
    std::cout << i << ' ';
```

**Ausgabe/Output:**

2 P

---

(b) 

```
unsigned int n = 32;
do {
    std::cout << n % 3 << ' ';
```

```
    n /= 2;
} while (n > 0);
```

**Ausgabe/Output:**

2 P

---

(c) 

```
double d = 1.0;
while (d != 0.125) {
    std::cout << d << ' ';
```

```
    d = d * 0.5;
}
```

**Ausgabe/Output:**

2 P

---

(d) 

```
for (int n = 0; n <= 10; n += 2) {
    if (n % 3 == 0)
        continue;
    std::cout << n << ' ';
```

```
}
```

**Ausgabe/Output:**

2 P

### 3 Primzahlzerlegung (6 Punkte)

Betrachten Sie das folgende Programm, das eine Funktion zur Primzahlzerlegung einer natürlichen Zahl definiert und ihre Benutzung an einem Beispiel zeigt. Ihre Aufgabe ist es, die Definition der Funktion `prime_factors` so zu ergänzen, dass sich eine korrekte Implementierung ergibt.

*Consider the following program that defines a function for computing the prime factorization of a natural number and illustrates its usage with an example. Your task is to complete the function `prime_factors` in such a way that a correct implementation is obtained.*

---

```
#include<iostream>
#include<cassert>

// PRE: n >= 2
// POST: outputs the prime factorization of n as a list of prime factors
//       in nondecreasing order, and with the correct multiplicities
void prime_factors (unsigned int n)
{
    assert (n >= 2);
    unsigned int d = 2;
    while ( [REDACTED] ) {
        if ( [REDACTED] ) {
            std::cout << d << " ";
            [REDACTED];
        } else {
            [REDACTED];
        }
    }
}

int main()
{
    prime_factors (24); // 2 2 2 3
    return 0;
}
```

---

## 4 Zeiger und Bereiche (8 Punkte)

Betrachten Sie folgendes Programmskelett. Ihre Aufgabe ist es, eine korrekte Definition der Funktion `cyclic_shift` anzugeben. Diese Funktion verschiebt jedes Element in einem gegebenen Bereich um eine Position nach links, wobei das erste Element des Bereichs ans Ende verschoben wird. Aus der Folge (1, 2, 3, 4, 5) wird also die Folge (2, 3, 4, 5, 1). Falls der Bereich leer ist, hat der Funktionsaufruf keinen Effekt. **Hinweis:** Sie können zum Beispiel wie folgt vorgehen: speichern Sie das erste Element des Bereichs; durchlaufen Sie den Bereich vom ersten bis zum vorletzten Element und ersetzen Sie jedes Element durch seinen Nachfolger; zum Schluss ersetzen Sie das letzte durch das gespeicherte Element.

---

```
#include<iostream>
// PRE: [begin, end) is a valid range, possibly empty
// POST: the sequence a_1, a_2, ..., a_n of numbers in the range
//       [begin, end) is replaced by the sequence a_2, ..., a_n, a_1
void cyclic_shift (int* begin, int* end) {

```

```

}
int main()
{
    int a[] = {1, 2, 3, 4, 5};
    cyclic_shift (a, a+5);
    for (int i=0; i<5; ++i) std::cout << a[i] << ' '; // 2 3 4 5 1
    return 0;
}
```

---

*Consider the program skeleton above. Your task is to provide a correct definition of the function `cyclic_shift`. This function moves every element in the range one place to the left, where the first element is moved to the back. Hence, the sequence (1, 2, 3, 4, 5) is transformed into the sequence (2, 3, 4, 5, 1). If the range is empty, the function call has no effect. **Hint:** you may for example proceed as follows: store the first element; iterate over the range, from the first to the second-to-last element, and replace each element with its successor. Finally, replace the last element by the stored element.*

## 5 Normalisierte Flieskommazahlen (8 Punkte)

Betrachten Sie die Funktion *Consider the function*

$$f(a, b, c, e) = (1 + a \cdot 2^{-1} + b \cdot 2^{-2} + c \cdot 2^{-3}) \cdot 2^e,$$

$$a, b, c \in \{0, 1\}, \quad e \in \mathbb{Z}.$$

Geben Sie in den folgenden Gleichungen jeweils Werte für die Argumente  $a, b, c$  und  $e$  so an, dass der Funktionswert die angegebene reelle Zahl ist!

*In each of the following equalities, supply values for the arguments  $a, b, c$  and  $e$  such that the function value is the required real number!*

$$(a) \left( 1 + \boxed{\phantom{0}} \cdot 2^{-1} + \boxed{\phantom{0}} \cdot 2^{-2} + \boxed{\phantom{0}} \cdot 2^{-3} \right) \cdot 2^{\boxed{\phantom{0}}} = 2$$

$$(b) \left( 1 + \boxed{\phantom{0}} \cdot 2^{-1} + \boxed{\phantom{0}} \cdot 2^{-2} + \boxed{\phantom{0}} \cdot 2^{-3} \right) \cdot 2^{\boxed{\phantom{0}}} = \frac{7}{8}$$

$$(c) \left( 1 + \boxed{\phantom{0}} \cdot 2^{-1} + \boxed{\phantom{0}} \cdot 2^{-2} + \boxed{\phantom{0}} \cdot 2^{-3} \right) \cdot 2^{\boxed{\phantom{0}}} = 30$$

$$(d) \left( 1 + \boxed{\phantom{0}} \cdot 2^{-1} + \boxed{\phantom{0}} \cdot 2^{-2} + \boxed{\phantom{0}} \cdot 2^{-3} \right) \cdot 2^{\boxed{\phantom{0}}} = 4.5$$

## 6 Rekursion (8 Punkte)

Betrachten Sie die folgenden vier Funktionen g, h, i und j. Beantworten Sie die Fragen auf der nächsten Seite! *Consider the following four functions g, h, i and j. Answer the questions on the next page!*

---

```
#include<iostream>

unsigned int g (unsigned int n)
{
    if (n == 0)
        return 0;
    else
        return g (n / 10) + n % 10;
}

bool i (unsigned int n);
bool h (unsigned int n)
{
    if (n == 0)
        return true;
    else
        return i (n-1);
}

bool i (unsigned int n)
{
    if (n == 0)
        return false;
    else
        return h (n-1);
}

void j (unsigned int n, unsigned int& k)
{
    if (n == 0)
        ++k;
    else
        for (unsigned int i=0; i<n; ++i)
            j (n-1, k);
}
```

- 
- (a) Betrachten Sie folgendes Hauptprogramm und geben Sie für jede Ausgabe den korrekten Wert an! *Consider the following main program and provide for each output the correct value!* 4 P
- 

```
int main()
{
    std::cout << g (57) << '\n';    // output: 
    std::cout << h (4) << '\n';    // output: 
    std::cout << i (4) << '\n';    // output: 
    unsigned int k = 0;
    j (2, k);
    std::cout << k << '\n';        // output: 

    return 0;
}
```

---

- 
- (b) Geben Sie für jede der vier Funktionen eine Nachbedingung an, die das Verhalten der Funktion vollständig beschreibt! *For each of the four functions, write down a postcondition that completely describes the behavior of the function!* 4 P
- 

```
// POST: 
unsigned int g (unsigned int n);
```

```
// POST: 
bool h (unsigned int n);
```

```
// POST: 
bool i (unsigned int n);
```

```
// POST: 
void j (unsigned int n, unsigned int& k);
```

---

## 7 EBNF (7 Punkte)

Die Kaffeehauskette Barstucks bietet diverse Espressogetränke an, die durch folgende EBNF beschrieben werden. Beantworten Sie die Fragen unten und auf der nächsten Seite!

**Anmerkung:** Zusätzliche Leerschläge sind im Rahmen der EBNF bedeutungslos.

---

```
beverage = size [milk] [temperature] product ["with" digit "shots"]
size = "Tall" | "Grande" | "Venti"
milk = "soy" | "nonfat" | "lowfat"
temperature = "iced" | {"extra"} "hot"
product = "Americano" | "Mocha" | "Latte" | "Cappucino"
digit = "1" | "2" | "3" | "4"
```

---

*The coffee house chain Barstucks is offering a variety of espresso beverages, described by the EBNF above. Answer the questions below and on the next page!*

**Remark:** *Additional whitespaces are irrelevant in the context of this EBNF.*

---

(a) Wahr oder falsch? *true or false?*

1 P

Die folgende Zeichenkette ist ein gültiges Getränk (beverage) nach der EBNF:

*The following string is a valid beverage (beverage) according to the EBNF:*

Tall hot Cappucino with 2 shots

---

(b) Wahr oder falsch? *true or false?*

1 P

Die folgende Zeichenkette ist ein gültiges Getränk (beverage) nach der EBNF:

*The following string is a valid beverage (beverage) according to the EBNF:*

Venti iced soy Americano

---

(c) Wahr oder falsch? *true or false?*

1 P

Die folgende Zeichenkette ist ein gültiges Getränk (beverage) nach der EBNF:

*The following string is a valid beverage (beverage) according to the EBNF:*

Grande extra extra extra hot Mocha with 4 shots

---

(d) Wahr oder falsch? *true or false?*

1 P

Die folgende Zeichenkette ist ein gültiges Getränk (beverage) nach der EBNF:

*The following string is a valid beverage (beverage) according to the EBNF:*

Venti Cappucino with 5 shots

---

(e) Wahr oder falsch? *true or false?*

1 P

Die folgende Zeichenkette ist ein gültiges Getränk (beverage) nach der EBNF:

*The following string is a valid beverage (beverage) according to the EBNF:*

Venti nonfat extra hot Latte with 1 shot

---

(f) Wie muss die Regel für milk abgeändert werden, damit optional noch der Fettgehalt der Low-Fat-Milch angegeben werden kann (nicht derjenige der Soja- oder No-Fat-Milch). Der Fettgehalt ist entweder 1% oder 2%, und eine Beispielspezifikation ist "lowfat 1%"? Geben Sie die neue Regel an!

2 P

*How do you have to change the rule for milk in order to optionally specify the fat content of the low fat milk (not the one of the soy or the no fat milk). The fat content may either be 1% or 2%, and an example specification is "lowfat 1%"? Write down the new rule!*

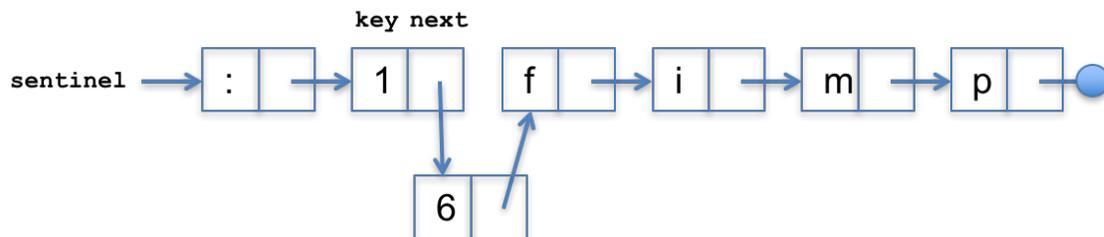
## 8 Sortierte Liste (8 Punkte)

Eine sortierte Liste ist eine verkettete Liste, bei der die Elemente in sortierter Reihenfolge gespeichert sind; in unserem Fall sind die Elemente Zeichen, aufsteigend sortiert nach ASCII code, wobei wir ein künstliches erstes Zeichen ':' (Wächter) speichern, um beim Einfügen in die Liste Spezialfälle zu vermeiden. *A sorted list is a linked list, storing its elements in sorted order; in our case, the elements are characters, sorted by increasing ASCII code, where we store an artificial first element ':' (sentinel) to avoid special cases during insertion.*



Ihre Aufgabe ist es, die Definitionen zweier Memberfunktionen der Klasse `sorted_list` korrekt zu ergänzen. Funktion `insert` fügt ein neues Zeichen hinzu, wobei die sortierte Reihenfolge aufrechterhalten wird. Im Beispiel führt das Einfügen des Zeichens '6' zu folgender Liste.

*Your task is to correctly complete the definitions of two member functions of the class `sorted_list`. Function `insert` adds a new character, maintaining the sorted order. In the example, inserting the character '6' leads to the following list.*



Funktion `print` gibt die Elemente der Liste in der sortierten Reihenfolge aus. Nach Einfügen von '6' wie oben gezeigt erhalten wir als Ausgabe `16fimp`. Ergänzen Sie das folgende Programm, so dass sich eine korrekte Implementierung ergibt! *Function `print` outputs the elements of the list in sorted order. After insertion of '6' as shown above, the output will be `16fimp`. Complete the following program so that you get a correct implementation!*

```
#include<iostream>

struct list_node
{
    list_node (char k, list_node* n)
        : key (k), next (n)
    {}
    char key;
    list_node* next; // 0 means: no next element
};
```

```

class sorted_list
{
public:
    // POST: generates an empty list
    sorted_list()
        : sentinel (new list_node (':', 0)) // sentinel
    {}

    // POST: inserts k into *this so that *this remains sorted by increasing ASCII code
    void insert (char k)
    {
        list_node* p = sentinel;
        while ( [redacted] && [redacted] )
            [redacted];
        p->next = [redacted];
    }

    // POST: prints all elements of *this, sorted by increasing ASCII code
    void print () const
    {
        for (list_node* p = [redacted]; [redacted]; [redacted])
            [redacted];
        std::cout << '\n';
    }
private:
    list_node* sentinel;
};

int main()
{
    sorted_list l;
    std::string s ("ifmp16");
    for (int i = 0; i < s.length(); ++i)
        l.insert (s[i]);
    l.print (); // 16fimp
    return 0;
}

```