

Prüfung **(Lösung)**
Informatik I (D-ITET)
 Felix Friedrich, Martin Bättig
 ETH Zürich, 10.8.2018.

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 von Ihnen gesprochene Sprachen).
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!
5. Störungen durch irgendjemanden oder irgendetwas melden Sie bitte sofort der Aufsichtsperson.
6. 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.
7. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen.
8. 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 languages spoken by yourself).*
- 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!*
- 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.*

Question:	1	2	3	4	5	6	7	Total
Points:	6	8	7	7	10	12	10	60
Score:								

Aufgabe 1: Typen und Werte (6P)

Geben Sie für jeden der Ausdrücke jeweils C++-Typ und Wert an. *For each of the expressions provide C++ type and value.*

/1P (a) `false && false || true && true`

Typ / *Type*

`bool`

Wert / *Value*

`true | 1`

/1P (b) `3 / 4 * 4`

Typ / *Type*

`int`

Wert / *Value*

`0`

/1P (c) `489 % 7 + 489 / 7 * 7`

Typ / *Type*

`int`

Wert / *Value*

`489`

/1P (d) `0xaffe + 0x2 == 0xb000`

Typ / *Type*

`bool`

Wert / *Value*

`true | 1`

/1P (e) `10.0 + 1 / 2`

Typ / *Type*

`double`

Wert / *Value*

`10.0`

/1P (f) `'b' + 'A' - 'B'`

Typ / *Type*

`char`

Wert / *Value*

`'a'`

Aufgabe 2: Konstrukte (8P)

Geben Sie zu folgenden Codestücken jeweils die erzeugte Ausgabe an.

Provide the output for each of the following pieces of code.

(a)

```
int a = 1;
int* b = &a;
b = b + 100;
std::cout << 2 * a;
```

Ausgabe / *Output*: 2

/2P

(b)

```
int s = 0;
for (int i = 0; i < 10; s += ++i){}
std::cout << s;
```

Ausgabe / *Output*: 55

/3P

(c)

```
int val = 0x111;
while (val > 0){
    std::cout << val % 2;
    val /= 2;
}
```

Ausgabe / *Output*: 100010001

/3P

Aufgabe 3: EBNF (7P)

Die folgende EBNF definiert sehr einfache Ausdrücke (z.B. zum Suchen in Texten). Beantworten Sie die nachfolgenden Fragen.

Anmerkung: Leerschläge sind im Rahmen der EBNF bedeutungslos.

The following EBNF defines very simple expressions (e.g. to search in texts). Answer the questions below.

Remark: Whitespaces are irrelevant in the context of this EBNF.

```
expression = term { term }.
term       = factor [quantor].
factor     = letter | "[" letter "-" letter "]" | "(" expression)".
quantor    = "*" | "?" | "+".
letter     = "a" | "b" | "c" | "d" | "e" | "t" | "h".
```

/5P

(a) Geben Sie in folgender Matrix an, ob die Zeichenkette ein Ausdruck (expression) gemäss der EBNF entspricht oder nicht. Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

Provide in the following matrix if the string corresponds to a valid expression according to the EBNF. Correct answers yield 1 point, wrong answers yield -1 point.

Zeichenkette <i>String</i>	gültig <i>valid</i>	ungültig <i>invalid</i>
[a-c]*	x	
c++		x
(abc)()?		x
a?b+c*(d?)*	x	
(eth)*	x	

/2P

(b) Ändern Sie genau eine Produktionsregel der EBNF ab, so dass die folgenden vier Zeichenketten (ohne Gänsefüsschen) jeweils gültige Expressions sind.

Modify one and only one production rule of the EBNF such that the following four expressions (without the quotes) are valid expressions each.

"(a|b)*" "abc|cde" "a?|b*|c+" "(ab)*|(cd)+"

Geänderte Zeile / *Modified line:*

```
term = factor [quantor] {"|" factor [quantor]}. or
expression = term { ["|"] term }
```

Aufgabe 4: Fließkommasyeme (7P)

- (a) Beantworten Sie die folgende Fragen und nehmen Sie dafür an, dass Flieskommazahlen sich gemäss dem IEEE-Standard 754 für Flieskommazahlen-Arithmetik verhalten! Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

Answer the following questions, assuming that floating point numbers behave according to the IEEE 754 standard for floating point arithmetics! Correct answers yield 1 point, wrong answers yield -1 point.

/3P

Frage Question	wahr <i>true</i>	falsch <i>false</i>
Die Literale 0.1 und 0.1f haben gleich viele signifikante Stellen. <i>Literals 0.1 and 0.1f have the same number of significant digits.</i>		x
Es gibt ein Wert a vom Typ float für den gilt: $a + 1 == a$. <i>There exists a value a of type float for that holds: $a + 1 == a$.</i>	x	
Jede 32-bit Zahl vom Typ int kann ohne Wertänderung in den Typ float konvertiert werden. <i>Each 32-bit number of type int is convertible into the type float without change in value.</i>		x

- (b) Beantworten Sie die folgende Fragen zum normalisierten Flieskommasytem $F^*(\beta, p, e_{min}, e_{max})$ mit $\beta = 2, p = 5, e_{min} = -3, e_{max} = 3$. Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

Answer the following questions regarding the normalized floating point system $F^(\beta, p, e_{min}, e_{max})$ with $\beta = 2, p = 5, e_{min} = -3, e_{max} = 3$. Correct answers yield 1 point, wrong answers yield -1 point.*

/4P

Frage Question	wahr <i>true</i>	falsch <i>false</i>
Das Flieskommasytem F^* kann die Zahl 0 repräsentieren. <i>The floating point system F^* can represent the number 0.</i>		x
15 ist die grösste Zahl, welche im Flieskommasytem F^* repräsentierbar ist. <i>15 is the biggest number representable in floating point system F^*</i>		x
1/16 ist im Flieskommasytem F^* exakt repräsentierbar. <i>1/16 is exactly representable in the floating point system F^*</i>		x
11 und 11.5 folgen im Flieskommasytem F^* direkt aufeinander, d.h., es gibt keine Zahl $Z \in F^*$ für die gilt $Z > 11$ und $Z < 11.5$. <i>In F^*, 11 and 11.5 follow each other without a gap, i.e., there is no number $Z \in F^*$, where $Z > 11$ and $Z < 11.5$.</i>	x	

Aufgabe 5: Rekursion (Pascalsches Dreieck) (10P)

Im folgenden ist das **Pascalsche Dreieck** p (in einer Matrixvariante) abgebildet.

Der Eintrag $p(1, 1)$ in Zeile $r = 1$ und Spalte $c = 1$ hat den Wert 1: $p(1, 1) = 1$.

Darüber hinaus gilt $p(r, c) = 0$ für alle $c \leq 0$ oder für alle $r \leq 0$.

Jeder andere Wert an Zeile r und Spalte c wird durch Addition der Zahlen darüber (Zeile $r-1$, Spalte c) und links davon (Zeile r , Spalte $c-1$) gebildet.

Below Pascal's Triangle p is shown (in a matrix variant).

The number $p(1, 1)$ in row $r = 1$ and column $c = 1$ has value 1: $p(1, 1) = 1$.

Moreover it holds that $p(r, c) = 0$ for all $c \leq 0$ or for all $r \leq 0$.

Every other value at row r and column c is provided as the value above, at row $r-1$ and column c , and left, row r and column $c-1$.

	Spalten / columns →					
Index	0	1	2	3	4	...
0	0	0	0	0	0	...
1	0	1	1	1	1	...
2	0	1	2	3	4	...
3	0	1	3	6	10	...
4	0	1	4	10	20	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Zeilen / rows ↓



- (a) Ergänzen Sie die folgende Funktion so, dass sie die Zahl $p(r, c)$ in Zeile r und Spalte c berechnet.

Complement the following method such that it computes the number $p(r, c)$ in row r and column c .

/6P

```
// Compute number in Pascal's triangle at row r and column c
int p(int r, int c){
    if (r <= 0 || c <= 0) [1p] {
        return 0;
    }
    if (c == 1 && r == 1) [or also works] [1p] {
        return 1;
    }
    return p(r-1,c) + p(r,c-1); [4p]
}
```

- (b) Geben Sie einen kurzen informellen Beweis dafür an, dass Ihre Methode für alle Eingaben (Parameter) $r \geq 0, c \geq 0$ terminiert.

Provide a short informal proof that your method will terminate for all inputs (parameters) $r \geq 0, c \geq 0$.

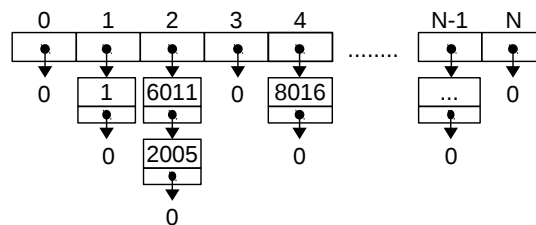
/4P

For each recursive call of p the r or c argument decrease. When r or c reach 0 the base case is reached and no further call takes place. [2p for decrease, 2p for mentioning the base case. 4p for any other correct complete explanation]

Aufgabe 6: Integer-Sets mit Hashing (12P)

Die dynamische Datenstruktur `HashSet` bildet eine Menge von Integerzahlen ab. Die Datenstruktur hat N Fächer. Ein Fach ist eine verkettete Liste (unsortiert) repräsentiert durch den Listenkopf (siehe Abbildung). Funktion `get_hash` berechnet einen Hash-Wert aus einer Zahl n , welcher n eindeutig einem Fach f zuordnet. Zahl n kann nur in der verketteten Liste des Fachs f und dort maximal einmal gespeichert werden (in diesem Fall ist sie Teil der Menge). Vervollständigen Sie die Implementation auf der rechten Seite.

The dynamic data structure `HashSet` implements a set of integer numbers. The data structure has N compartments. Each compartment is a linked list (unsorted) represented by its head node (see figure). The function `get_hash` calculates a hash value out of a number n that assigns n uniquely to a compartment f . Number n can only be stored in the linked list of compartment f and in this list maximal once (in which case n is part of the set). Complete the implementation of class `HashSet`.



```

struct Node {
    int value; Node* next;
    Node(int v, Node* n) : value(v), next(n) {};
};
class HashSet {
    const int N = 2003;
    Node* nodes[N] = { 0 }; // all initialized with 0
    int get_hash(int n) { return n % N; } // hash of integer n
public:
    bool contains(int n);
    void add(int n); void remove(int n);
};
// PRE: n:      gesuchter Wert          (value to be found)
//       node:  erster Listknoten      (first list node)
// POST: gefunden: Gibt true zurueck    (found: returns true)
//       node:  Knoten von n           (node containing n)
//       prev:  Vorgaenger oder 0 falls Kopf (predecessor or 0 if head)
//       nicht gefunden: Gibt false zurueck (not found: returns false)
//       node:  0
//       prev:  Endknoten oder 0, falls leer (tail node or 0 if empty)
bool find(Node*& prev, Node*& node, int n) {
    prev = 0;
    while(node != 0 && node->value != n) {
        prev = node; node = node->next;
    }
    return node != 0;
}

```

- (a) Vervollständigen Sie die Funktion `contains` *Complete the function `contains` according to its post condition.*

/2P

```
// POST: Gibt true zurueck, falls n Element der Menge ist (sonst false)
//       (returns true if n is an element of the set, otherwise false)
bool HashSet::contains(int n) {
    Node* node = nodes[get_hash(n)];
    Node* prev;

    return find(prev, node, n);
}
```

- (b) Vervollständigen Sie die Funktion `add` *Complete the function `add` according to its post condition.*

/4P

```
// POST: n ist Element der Menge (n is an element of the set)
void HashSet::add(int n) {
    if (!contains(n)) {
        // fuege n als erstes Element hinzu (add n as first element)
        int hash = get_hash(n);

        Node* new_node = new Node(n, nodes[hash]);

        nodes[hash] = new_node;
    }
}
```

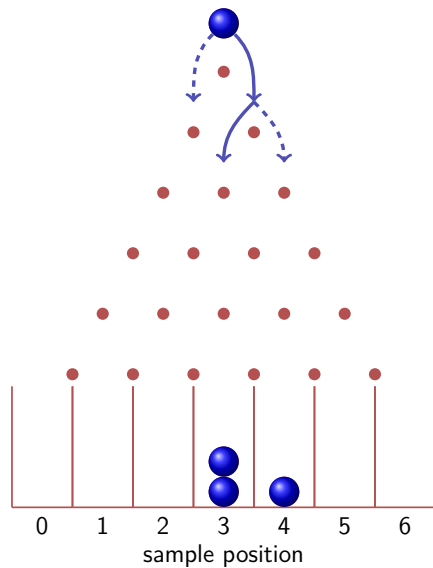
- (c) Vervollständigen Sie die Funktion `remove` *Complete the function `remove` according to its post condition.*

/6P

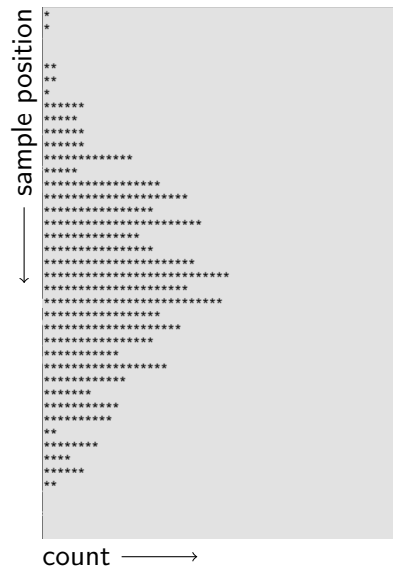
```
// POST: n ist kein Element der Menge (n is not an element of the set)
void HashSet::remove(int n) {
    int hash = get_hash(n);
    Node* node = nodes[hash];
    Node* prev;
    if (find(prev, node, n)) {
        if (prev == 0)
            nodes[hash] = nodes[hash]->next;
        else
            prev->next = node->next;
        delete node;
    }
}
```

Aufgabe 7: Galton Brett (10P)

Ziel dieser Aufgabe ist die Simulation eines Galton Bretts. Beim Galton Brett fällt ein Kugel auf einen Nagel, an welchem sie sich mit gleicher Wahrscheinlichkeit nach links oder rechts weiterbewegt. Danach fällt sie erneut auf einen Nagel, wo dasselbe passiert. Nach n Nagelzeilen fällt die Kugel in eines von $n + 1$ Fächern (Abbildung links).



Goal of this task is to simulate a Galton board. On the Galton board, a ball falls on a nail where it continues to move to the left or to the right with equal probability. Then it falls on the next nail where the same happens. After n rows of nails the ball falls into one of $n + 1$ boxes (cf. left figure below).



/4P (a) Vervollständigen Sie die Funktion `simulate`, so dass sie das Fallen einer Kugel durch ein Galton-Brett mit n Zeilen simuliert. Die gegebene Funktion `ran` gibt einen der beiden Werte 0 oder 1 mit gleicher Wahrscheinlichkeit zurück. Tipp: wie oft muss die Kugel nach rechts fallen, um in Topf m zu landen ($0 \leq m \leq n$)?

Complement the function `simulate` such that the moving of a ball through a Galton board with n rows is simulated. The given function `ran` returns of the values 0 or 1 with equal probability. Hint: how often does the ball have to go right in order to hit box m ($0 \leq m \leq n$)?

/4P (b) Vervollständigen Sie die `histogram` Funktion so, dass ein Histogramm angezeigt wird. Das Histogramm zeigt einen Ausschnitt eines Vektors, welcher durch Iteratoren `left` und `right` spezifiziert ist. Bei dem Histogramm wird die Zahl zu jedem Eintrag des Vektors durch die entsprechende Anzahl Sterne in der jeweiligen Zeile dargestellt (rechte Abbildung).

Complement the `histogram` function such that a histogram is displayed. The histogram should show a part of the vector that is specified by iterators `left` and `right`. The histogram shows each entry of the vector by the corresponding number of stars in the corresponding line (cf. right figure above).

/2P (c) Vervollständigen Sie die `main` Funktion so, dass das Histogramm des Vektors `v` zwischen den durch `left` and `right` bezeichneten Indexp Grenzen angezeigt wird.

Complement the `main` function such that the histogram of the vector `v` is shown from the indices given by `left` and `right`.

```
typedef std::vector<int>::const_iterator iterator;
// return value 0 or 1 with probability 1/2 each
int ran(); // implementation omitted for brevity

// pre: n >= 0
// post: return sample position 0 <= pos <= n
int simulate(int n){
    int pos = 0;
    for (int i = 0 ; i < n ; ++i){
        pos += ran();
    }
    return pos ;
}

// pre: valid range left .. right
// post: show histogram of values between left and right
void histogram(const iterator left, const iterator right){
    for ( iterator s = left; s != right; ++s ){
        for ( int i = 0; i<*s; ++i ){
            std::cout << "*";
        }
        std::cout << std::endl;
    }
}

int main(){
    int height = 200; // height of the galton board
    int left = 80; // left bounds of the histogram (inklusive)
    int right = 120; // right bounds of the histogram (inklusive)
    int samples = 400; // number of balls
    std::vector<int> v(height+1);
    for (int i = 1; i<=samples; ++i)
        v[simulate(height)]++;

    histogram( v.begin()+left, v.begin()+right+1 );
    return 0;
}
```
