Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prüfung

Informatik I (D-ITET)

Felix Friedrich, Martin Bättig

ETH Zürich, 9.2.2018.

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:

1. Dauer der Prüfung: 60 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für 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 bitte deutlich durchstreichen! Korrekturen bei Multiple-Choice Aufgaben unmissverständlich anbringen!
5. Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies 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. Es darf zur gleichen Zeit immer nur eine Studentin oder ein Student zur Toilette.
8. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

General guidelines:

1. *Exam duration: 60 minutes.*
2. *Permitted examination aids: dictionary (for spoken languages).*
3. *Use a pen (black or blue), not a pencil. Please write legibly. We will only correct solutions that we can read.*
4. *All solutions must be written directly onto the exercise sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be crossed out clearly. Corrections to answers of multiple choice questions must be provided without any ambiguity.*
5. *If you feel disturbed by anyone or anything, immediately let the supervisor of the exam know this.*
6. *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 preliminarily is only possible until 15 minutes before the exam ends.*
7. *If you need to go to the toilet, raise your hand and wait for a supervisor. Only one student can go to the toilet at a time.*
8. *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	8	10	10	8	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) `1 / 2 * 2.0`

Typ / *Type* | Wert / *Value*

/1P (b) `12 - 7 % 5`

Typ / *Type* | Wert / *Value*

/1P (c) `true && ! true == false || false`

Typ / *Type* | Wert / *Value*

/1P (d) `0x10 + 0x12 == 32`

Typ / *Type* | Wert / *Value*

/1P (e) `3 / 2.0f + 10 / 2.0`

Typ / *Type* | Wert / *Value*

/1P (f) `4u - 5 < 0`

Typ / *Type* | Wert / *Value*

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 = 10;
int& b = a;
b -= a;
std::cout << a + b;
```

Ausgabe / *Output*: /2P

(b)

```
int a[] = {1,8,0,2,7,4,3,6,5};
for (int i = 3; a[i] != 5; i = a[i]){
    std::cout << *(a+i);
}
```

Ausgabe / *Output*: /3P

(c)

```
int res = 0;
int d = 1;
int val = 101010;
while (val > 0){
    res += val % 10 * d;
    d *= 2;
    val /= 10;
}
std::cout << res;
```

Ausgabe / *Output*: /3P

Aufgabe 3: EBNF (8P)

Die folgende EBNF definiert gewisse Ausdrücke in einer Prefixschreibweise.
Beantworten Sie die nachfolgenden Fragen.

Anmerkung: Leerschläge sind im Rahmen der EBNF bedeutungslos.

The following EBNF defines certain expressions in a prefix notation.
Answer the questions below.

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

```
Expression = Operator Operand ',' Operand.
Operator   = '+' | '-' | '/' | '*'.
Operand    = Number | Expression.
Number     = Integer | Integer '.' Integer.
Integer    = Digit {Digit}.
Digit      = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' .
```

/6P (a) Geben Sie in folgender Matrix an, ob die Zeichenkette einem gültigen Ausdruck (Expression) gemäss der EBNF entspricht oder nicht. Falsche Antworten ergeben keine negativen Punkte.

Provide in the following matrix if the string corresponds to a valid expression according to the EBNF. Wrong answers do not result in negative points.

Zeichenkette *String*	gültig *valid*	ungültig *invalid*
+ + 3 , 5 , 6		
+ (3.5 , 2)		
+ * 3.3 , + 5 , 6 , 2.2, 1.1		
3 - * 5 , 6		
+ * 3 , 5 , * 6.2 , 0.7		
+ 3.7 - 4.7		

/2P (b) Ändern Sie genau eine Produktionsregel der EBNF ab, so dass folgender Ausdruck (expression) gültig wird.

Modify one and only one production rule of the EBNF such that the following expression becomes valid.

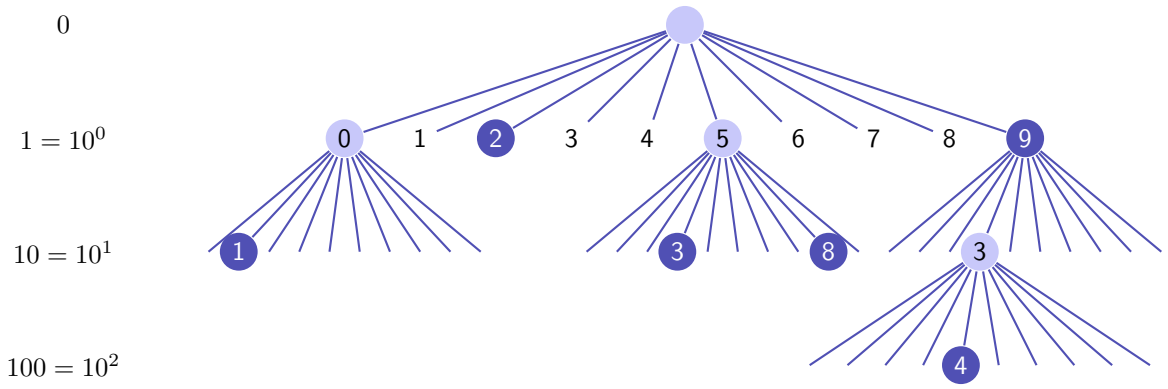
```
( ( 3.5 + 6) * ( 2 + 7.2 ) )
```

Geänderte Zeile / *Modified line*:

Aufgabe 4: Dynamische Datenstrukturen (Tries) (10P)

In dieser Aufgabe betrachten wir einen Trie, eine Spezialform eines Baumes zum schnellen Finden von Werten in einer Menge. Der in dieser Aufgabe vorgestellte Baum hat die Ordnung 10: jeder Knoten hat 10 Kinder.

In this task we consider a trie, a special form of a tree to quickly find integer values within a set. The tree proposed in this task has an order of 10: every node has 10 children.



Unser Baum speichert beliebige positive ganze Zahlen. Jede Ebene des Baumes (ausser der Wurzel) steht für eine Zehnerpotenz. Die Zehnerpotenz ist auf der linken Seite des Baumes dargestellt. Eine Zahl ist dann im Baum enthalten, wenn die Knoten Ihrer Darstellung als Dezimalzahl im Baum enthalten sind und der Blattknoten der Zahl markiert ist (im Graphen dargestellt durch dunkle Einfärbung). Der Baum in obiger Abbildung enthält so die Zahlen 10, 2, 35, 85, 9 und 439.

Auf den nächsten Seite sehen Sie die Definitionen der Klassen TrieNode und Trie. Beantworten Sie die folgende Fragen.

Es geht in dieser Aufgabe nicht um korrektes Aufräumen des Speichers (kein delete, keine Dreierregel)

(bitte blättern)

Our tree can store arbitrary positive integers. Every level of the tree (besides the roots) stands for a power of ten. The powers of ten are displayed on the left of the tree. A number is contained in the tree if the nodes of its representation as decimal number are contained in the tree and if the leaf node of a number is marked (represented with a dark background). The tree in the figure above contains the numbers 10, 2, 35, 85, 9 *and* 439.

Further below and on the next page you find the definition of the classes TrieNode and Trie. Answer the following questions.

This task is not about correct cleaning up of allocated memory (no delete, no rule of three)

(please turn pages)

/3P (a) Wir nehmen an, dass die Methode `insert` der Klasse `Trie` korrekt implementiert ist gemäss ihrer Vor- und Nachbedingungen. Was ist dann die Ausgabe des folgenden Codestückes? Tipp: zeichnen Sie sich den entstehenden Baum kurz auf.

We assume that the method `insert` of the class `Trie` is implemented correctly according to the pre- and post conditions. What is then the output of the following piece of code? Hint: draw the arising tree shortly.

```
Trie t;
t.insert(10); t.insert(20); t.insert(231);
t.insert(2300); t.insert(4000); t.insert(0);
t.print();
```

Ausgabe / *Output*

/3P (b) Vervollständigen Sie den Code der Methode `insert` der Klasse Trie, so dass die Zahl in den Trie eingefügt wird, dass also alle benötigten Knoten alloziert sind und dass der zum Wert gehörige Knoten markiert ist.

Complement the code of the method `insert` of the class Trie such that the number is inserted in the Trie, i.e. such that all required nodes are allocated and such that the node corresponding to the value is marked.

/4P (c) Vervollständigen Sie den Code der Methode `has` der Klasse Trie, so dass sie korrekt zurückgibt, ob die angegebene Zahl im Trie enthalten ist.

Complement the code of the method `has` of the class Trie such that it returns correctly if the Trie contains the number provided.

```
// struct TrieNode only to be used within class Trie
struct TrieNode{
    std::vector<TrieNode*> children;
    bool marked;
    // constructor: unmarked node containing a vector of empty nodes
    TrieNode():children(std::vector<TrieNode*>(10)), marked(false){}
};

// class Trie to store positive integer numbers
class Trie{
    TrieNode* root; // invariant: root != nullptr
public:
    // constructor
    Trie(): root(new TrieNode()){}
```

```
    // recursive helper function for print()
    void printNode(TrieNode* n, int value, int d){
        if (n == 0){ return; }
        if (n->marked){ std::cout << value << " "; }
        for (int i = 0; i<10; ++i){
            printNode(n->children[i], value+d*i, 10*d);
        }
    }
    // output all numbers stored in the trie
    void print(){
        printNode(root, 0, 1);
    }
    // pre: value >= 0
    // post: all nodes that are required for representing value allocated,
    //       and the corresponding node marked
    void insert(int value){
        TrieNode* n = root;
        while (value > 0){
            if (                                                    ){
               n->children[value % 10] = new TrieNode();
            }
            n = n->children[value % 10];
            value = value / 10;
        }
                                                                    
    }
    // pre: value >= 0
    // post: return if value is contained in trie
    bool has(int value){
        TrieNode* n = root;
        while (n != 0 &&                    ){
            n =                                                     ;
            value = value / 10;
        }
        return                                                      ;
    }
};
```

Aufgabe 5: Klassen und Operator Overloading (10P)

Auf der rechten Seite sehen Sie die Implementation einer Klasse für Zeit. Unten auf dieser Seite sehen Sie, wie die Klasse benutzt werden soll. Implementieren Sie die fehlenden Teile wie im folgenden beschrieben.

On the right hand side you see the implementation of a class for time. Below on this page you see how the class should be used. Implement the missing parts as described in the following.

/3P (a) Implementieren Sie den Operator == so, dass er true zurückgibt, wenn zwei Zeiten gleich sind. Gleich sind hier zwei Zeiten, wenn sie die gleiche Tageszeit repräsentieren. In diesem Sinne sind die Zeiten 24:05:59 und 0:05:59 gleich.

Implement the operator == such that it returns true when the two times equal. Two times are equal when they represent the same time of the day. In this sense, 24:05:59 equals 0:05:59.

/1P (b) Implementieren Sie den Operator != so, dass er genau dann true zurückgibt, wenn die zwei Zeiten nicht gleich sind.

Implement the operator != such that it returns true if and only if the times to be compared are not equal.

/3P (c) Implementieren Sie den arithmetischen Zuweisungsoperator += so, dass Zeilen 5 und 6 des Codes der Main-Funktion erwartungsgemäss funktionieren

Implement the arithmetic assignment operator += such that lines 5 and 6 of the main function below works as expected.

/3P (d) Implementieren Sie den Ausgabeoperator << so, dass Zeile 7 des Codes der Main-Funktion erwartungsgemäss funktioniert.

Implement the output operator << such that line 7 of the code of the main function works as expected.

```
1 int main(){
2     Time second (0,0,1);
3     Time examStart (14,04,58);
4     Time examDuration (01,02,05);
5     Time examEnd = examStart + examDuration;
6     for (Time t=examStart; t != examEnd; t += second){
7         std::cout << t << std::endl;
8     }
9     // output:
10    // 14:4:58
11    // 14:4:59
12    // ...
13    // 15:7:2
14 }
```

```
class Time{
    int sec; // total number of seconds

    public:
    // construct time from hours, minutes and seconds
    Time (int h, int m, int s): sec(s+m*60+h*3600){}
    // add t to this time
    [          ] operator+= (const Time& t){
        [                                    ]
    }
    // two times match when they represent the same time during a day
    // Example: 0:0:5 must match 24:0:5
    bool operator==[                                ]{
        [                                                ];
    }
    // write time to stream out
    void write (std::ostream& out) const {
        out << sec / 3600 % 24 << ":" << sec / 60 % 60 << ":" << sec % 60;
    }
};
// output time on stream o
[                    ] operator<< (std::ostream& o, const Time t){
    [                                    ]
}
// return if times are not equal
bool operator !=(const Time& t1, const Time& t2){
    [                                                ];
}
// add time t1 and time t2
Time operator+(Time t1, const Time & t2){
    return t1 += t2;
}
```

Aufgabe 6: Unfair Würfeln (8P)

Sie wollen einen unfairen Würfel mit n Seiten simulieren.

We want to simulate an unfair dice with n faces.

/4P (a) Für die Wahrscheinlichkeit p_i für das Auftreten der Seite i gilt

For the probability p_i for the occurence of a face i it holds that

$$0 \leq p_i \leq 1 \quad (0 \leq i < n).$$

Für den Vektor $p = (p_i)_{0 \leq i < n}$ (die Verteilung) gilt zudem

Moreover, for the vector $p = (p_i)_{0 \leq i < n}$ (the distribution) it holds that

$$\sum_{i=0}^{n-1} p_i = 1$$

Vervollständigen Sie die Funktion `checkDistribution`, die true zurückgibt genau dann, wenn ein Vektor p diese beiden (Un)gleichungen erfüllt.

Complement the function `checkDistribution` that returns true if and only if a vector p fulfills these two (in)equalities.

/4P (b) Für das Ziehen von Zufallszahlen nach der Verteilung p zieht man mit einem vorhandenen Zufallszahlengenerator eine Zufallszahl v gleichverteilt aus dem Intervall $[0, 1)$. Aus dieser Zahl v erzeugt man dann eine ganze Zahl $0 \leq S(p, v) < n$ mit folgender Regel

In order to draw a random number according to the distribution p, using an existing random number generator, a number v is drawn uniformly distributed from the interval $[0, 1)$. From this number v an integer $0 \leq S(p, v) < n$ is generated according to the following rule

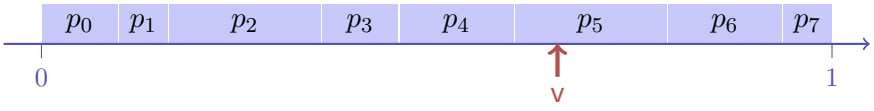
$$S(p, v) = \min\{0 \leq i < n : \sum_{k=0}^{i} p_k > v\}$$

Vervollständigen Sie die Funktion `sample`, so dass sie zu gegebenem Vektor p und Zahl $0 \leq v < 1$ den Wert $S(p, v)$ zurückgibt.

Complement the function `sample` such that it returns the value $S(p, v)$ for a given vector p and number $0 \leq v < 1$.

Die folgende Abbildung illustriert, wie $S(p, v)$ zu p und v gewählt wird. Hier: $n = 7$, $S(p, v) = 5$.

The following figure illustrates, how the value $S(p, v)$ for p and v is chosen. Here: $n = 7$, $S(p, v) = 5$.



```
typedef std::vector<double>::iterator Iterator;

// check distribution in the range [begin,end)
bool checkDistribution(Iterator begin, Iterator end){
    const double epsilon = 0.0000001; // assumed required floating point precision
    double sum = 0;
    for (Iterator p = begin;                                    ){
        if (                                                    ){
          return false;
        }
        sum += *begin;
        ++begin;
    }
                                                                
}

// sample from distribution p and sample value v
int sample (const std::vector<double>& p, double v){
    int res = 0;
    double sum = p[res];
    while(                                                      ){
        ++res;
                                                                ;
    }
    return res;
}

// return random value uniformly drawn from [0,1)
double Uniform(){
    ... // implementation omitted for brevity
}
// usage example
int main(){
    std::vector<double> p = {0.1, 0.2, 0.2, 0.4, 0.1};
    assert(checkDistribution(p.begin(), p.end()));
    double v = Uniform(); // random value in [0,1)
    int s = sample(p, v); // random number in {0,..,4}
}
```

Aufgabe 7: Zahlendarstellungen (10P)

/6P (a) Geben Sie ein möglichst knappes normalisiertes Fliesskommazahlensystem an, mit welchem sich die folgenden dezimalen Werte gerade noch genau darstellen lassen: jede Verkleinerung von p, $e_{\max}$ oder $-e_{\min}$ muss dazu führen, dass mindestens eine Zahl nicht mehr dargestellt werden kann.
Hinweis: p zählt auch die führende Ziffer.
Tipp: Schreiben Sie sich die normalisierte Binärzahldarstellung der Werte auf, wenn sie für Sie nicht offensichtlich ist.

Provide the smallest possible normalized floating point number system that can still represent the following values exactly: any decrease of the numbers p, $e_{\max}$ or $-e_{\min}$ must imply that at least one of the numbers cannot be represented any more.
Hint: p does also count the leading digit.
Tip: Write down the normalized binary representation of the values, if it is not obvious for you.

Werte / *Values*: 34, 1.75, $65/64$, $1/128$

$F^*(\beta, p, e_{\min}, e_{\max})$ mit / *with*
$\beta = 2$, $p =$, $e_{\min} =$, $e_{\max} =$.

/4P (b) Die Dezimalzahlen $x = 40$ und $y = 1$ sind im normalisierten Fliesskommasystem $F^*(2, 5, 0, 5)$ ($p = 5$) darstellbar.
Geben Sie das Resultat der Berechnung von $x + y$ in F^* an. Geben Sie die Antwort in dezimaler und binärer Darstellung an. Hinweis: Verwenden Sie arithmetisches Runden.
Geben Sie jeden einzelnen Schritt der Berechnung an.

The decimal numbers $x = 40$ and $y = 1$ are representable in the normalized floating point system $F^*(2, 5, 0, 5)$ ($p = 5$). Provide the result of $x + y$ when computed in F^*. Provide the answer in binary und decimal representation. Hint: Use arithmetic rounding to find this number. Provide each single step of the computation.