Informatik I **Prüfung 27. 1. 2016** ***B. Gärtner***

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! Für falsche Antworten werden keine negativen Punkte vergeben.
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. No negative points will be awarded for false answers.*
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.*

Aufgabe	1	2	3	4	5	6	7	8	$\sum$
Punkte									
Maximum	6	6	8	6	6	8	6	8	54

1 Typen und Werte *(6 Punkte)*

Geben Sie für jeden der drei Ausdrücke auf der rechten Seite jeweils C++-Typ und Wert an! Für den Typ `double` nehmen Sie den IEEE 754 Standard an!

Wir merken an, dass z.B. `integer` oder `boolean` keine C++-Typen sind und als falsche Antworten gewertet werden. Die korrekten Typen heissen `int` und `bool`.

Variablen `u`, `i` und `d` seien deklariert und initialisiert wie folgt.

```
unsigned int u = 5;
int i = -10;
double d = 0.25;
```

For each of the 3 expressions on the right, provide the C++ type and value!

Note that, for example, `integer` or `boolean` are not C++ types and will be considered as incorrect answers. The correct types are called `int` and `bool`. For the type `double`, assume the IEEE 754 Standard.

Variables `u`, `i` and `d` have been initialized as shown above.

(a) `u == 4 * ++d` 2 P

Typ/Type:

Wert/Value:

(b) `u / 3 * 2.0 + i` 2 P

Typ/Type:

Wert/Value:

(c) `u - (--i)--` 2 P

Typ/Type:

Wert/Value:

2 Schleifen *(6 Punkte)*

Wir betrachten ein Programm der Form

```
#include<iostream>
int main()
{
  // einfache Schleife:
  ...
  return 0;
}
```

Bestimmen Sie die Ausgabe des Programms für die drei einfachen Schleifen auf der rechten Seite!

We consider a program of the form

```
#include<iostream>
int main()
{
  // simple loop:
  ...
  return 0;
}
```

Determine the output of the program for the three simple loops on the right!

(a) Was gibt das Programm mit folgender einfachen Schleife aus? 2 P
What is the output of the program with the following simple loop?

```
double x = 2;
do {
  std::cout << x << " ";
  x /= 2;
} while (x > 0.1);
```

(b) Was gibt das Programm mit folgender einfachen Schleife aus? 2 P
What is the output of the program with the following simple loop?

```
bool x = false;
bool y = false;
while (!(x && y)) {
  std::cout << x << " " << y << " ";
  x = x || y;
  y = !y;
}
```

(c) Was gibt das Programm mit folgender einfachen Schleife aus? 2 P
What is the output of the program with the following simple loop?

```
for (int i = 10; --i > 1; i--)
  std::cout << i << " ";
```

3 Tribonacci-Zahlen *(8 Punkte)*

Betrachten Sie folgende Funktion. Beantworten Sie die Fragen auf der rechten Seite!

```
unsigned int trib (unsigned int n)
{
  std::cout << "*";
  if (n < 3) return n;
  return trib (n-1) + trib (n-3);
}
```

Consider the function above. Answer the questions on the right-hand side!

(a) Was ist der Wert des Funktionsaufrufs `trib(7)`?
What is the value of the function call `trib(7)`?

2 P

(b) Wie viele Sterne (*) gibt der Funktionsaufruf `trib(7)` aus?
How many stars (*) does the function call `trib(7)` output?

2 P

(c) Provide a definition of the following function!
Geben Sie eine Definition der folgenden Funktion an!

4 P

```
// POST: counts the number of stars that a call to trib (n) outputs
unsigned int stars_in_trib (unsigned int n)
{

}
```

4 Umgekehrte Polnische Notation *(6 Punkte)*

Die umgekehrte Polnische Notation (UPN) dient zum Aufschreiben arithmetischer Ausdrücke ohne Benutzung von Klammern. Dabei wird der Operator hinter die Operanden gesetzt. Der Ausdruck $(3 + 4) * 5$ wird in UPN zum Beispiel wie folgt notiert: `3 4 + 5 *`

Ein komplizierteres Beispiel ist $(3 + 4) * 5 - 13 * 8$:

```
3 4 + 5 * 13 8 * -
```

Gesucht ist eine BNF zur Beschreibung arithmetischer Ausdrücke mit den vier grundlegenden arithmetischen Operatoren `+, -, *, /` in UPN. Die Operanden dürfen beliebige natürliche Zahlen sein (führende Nullen erlaubt), müssen jedoch mit einem Leerzeichen enden, um sie vom folgenden Operanden oder Operator abzugrenzen. Die BNF soll die folgenden fünf Nichtterminale definieren. Die Definitionen von `operator, operand, digit` sind bereits vorgegeben; Ihre Aufgabe auf der rechten Seite ist es, die fehlenden zwei Nichtterminale zu definieren!

```
expression =
operator = "+" | "-" | "*" | "/"
operand = number " "
number =
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

Reverse Polish Notation (RPN) is used to write down arithmetic expressions without brackets. In RPN; the operand is put behind the operators. For example, the expression $(3 + 4) * 5$ is written in RPN as follows: `3 4 + 5 *`

A more complicated example is $(3 + 4) * 5 - 13 * 8$:

```
3 4 + 5 * 13 8 * -
```

We want to find a BNF that describes arithmetic expressions with the four basic arithmetic operators `+, -, *, /` in RPN. The operands can be arbitrary natural numbers (leading zeros allowed), but they must end with a blank in order to delimit them from the next operand or operator. The BNF should define the above five nonterminals. The definitions of `operator, operand, digit` are already given. Your task on the right-hand side is to define the missing two nonterminals!

(a) Geben Sie eine Definition des Nichtterminals `expression` an!
Provide a definition of the nonterminal `expression`!

3 P

(b) Geben Sie eine Definition des Nichtterminals `number` an!
Provide a definition of the nonterminal `number`!

3 P

5 Fehlersuche *(6 Punkte)*

Betrachten Sie das folgende fehlerhafte Programm mit Deklaration und Benutzung der Klasse `Clock`. Die Datenmember und Definitionen der Memberfunktionen sind für diese Aufgabe irrelevant und deshalb nicht angegeben. Beantworten Sie die Fragen auf der rechten Seite! Jede Zeilennummer darf nur einmal angegeben werden, auch wenn die Zeile mehrere (gleichartige) Fehler enthält.

```
1  class Clock {
2
3    // PRE: h < 24, m < 60, s < 60
4    // POST: creates a clock showing time h:m:s
5    Clock (unsigned int h, unsigned int m, unsigned int s);
6
7    // POST: advances the clock's time by one second
8    void tick () const;
9
10   // POST: h, m, s are set to the clock's hour, minute and second
11   void time (unsigned int h, unsigned int m, unsigned int s) const;
12
13 private:
14   // data members to store the time
15 };
16
17 main() {
18   Clock c = (22, 59, 59);
19   c.tick();
20
21   unsigned int h; unsigned int m; unsigned int s;
22   time (h, m, s); // 23:00:00
23
24   return 0;
25 }
```

The above program with declaration and usage of the class `Clock` contain some errors. The data members and definitions of the member functions are irrelevant for this assignment and are therefore omitted. Answer the questions on the right-hand side! Each line number can be specified only once, even if the line contains several errors (of the same type).

(a) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

(b) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

(c) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

(d) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

(e) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

(f) Geben Sie die Nummer einer fehlerhaften Zeile an und beschreiben Sie kurz den Fehler! 1 P
Specify the number of a line containing an error, and briefly describe the error!

A:

6 Lineare Algebra *(8 Punkte)*

Folgendes Programm deklariert und benutzt eine Klasse für dreidimensionale Vektoren; Vektoren können addiert und mit einem Skalar multipliziert werden. Für zwei Vektoren $v = (v_1, v_2, v_3)$, $w = (w_1, w_2, w_3)$ und eine reelle Zahl s gilt $v + w := (v_1 + w_1, v_2 + w_2, v_3 + w_3)$ und $sv := (sv_1, sv_2, sv_3)$. Ihre Aufgabe ist es, die Funktionen für Skalarmultiplikation und die Vektorausgabe zu definieren, wobei Sie die beiden gegebenen Konstruktoren benutzen dürfen. Beantworten Sie die Fragen auf der rechten Seite!

```
#include<iostream>

struct Vector {
  double x[3];

  // POST: creates the null vector
  Vector ();

  // POST: creates the vector (x1, x2, x3)
  Vector (double x1, double x2, double x3);
};

// POST: returns v + w (vector addition)
Vector operator+ (const Vector& v, const Vector& w);

// POST: returns s * v (scalar multiplication)
Vector operator* (double s, const Vector& v);

// POST: writes v to o in the format (v.x[0], v.x[1], v.x[2])
std::ostream& operator<< (std::ostream& o, const Vector& v);

int main()
{
  Vector v (1, 2, 3);
  Vector w (3, 4, 5);
  std::cout << 2 * v + 3 * w; // (11, 16, 21)
  return 0;
}
```

The program above declares and uses a class for threedimensional vectors. Vectors can be added and multiplied with a scalar. For two vectors $v = (v_1, v_2, v_3)$, $w = (w_1, w_2, w_3)$ and a real number s, we have $v+w := (v_1+w_1, v_2+w_2, v_3+w_3)$ and $sv := (sv_1, sv_2, sv_3)$. Your task is to define the functions for scalar multiplication, and vector output, where you may use the two given constructors. Answer the questions on the right-hand side!

(a) Definieren Sie die folgende Funktion zur Skalarmultiplikation! 3 P
Define the following function for scalar multiplication!

```
// POST: returns s * v (scalar multiplication)
Vector operator* (double s, const Vector& v) {
```

```

```

```
}
```

(b) Definieren Sie die folgende Funktion zur Ausgabe eines Vektors! Achten Sie darauf, dass Sie das in der Nachbedingung spezifizierte Format einhalten, das heisst den Vektor wie angegeben mit umschliessenden Klammern und kommaseparierten Elementen ausgeben! 5 P
Define the following function for vector output! Make sure that you comply with the format specified in the postcondition, i.e. that you output the vector using surrounding parentheses and comma-separated entries!

```
// POST: writes v to o in the format (v.x[0], v.x[1], v.x[2])
std::ostream& operator<< (std::ostream& o, const Vector& v) {
```

```

```

```
}
```

7 Reissverschluss *(6 Punkte)*

Seien $x = (x_1, x_2, \ldots, x_n)$ und $y = (y_1, y_2, \ldots, y_m)$ zwei Folgen mit möglicherweise unterschiedlicher Länge. Wir definieren den Reissverschluss von x und y als

$$\mathrm{zip}(x, y) = (x_1, y_1, x_2, y_2, \ldots, x_{\min(n,m)}, y_{\min(n,m)}).$$

Für $x = (1, 3, 5, 7, 9, 11)$ und $y = (2, 3, 4, 6, 8)$ gilt zum Beispiel $\mathrm{zip}(x, y) = (1, 2, 3, 4, 5, 6, 7, 8)$.

Gesucht ist eine Funktion, die zwei Folgen ganzer Zahlen x und y als Zeigerbereiche erhält und $\mathrm{zip}(x, y)$ ausgibt. Unten finden Sie das Skelett eines Programms, das eine solche Funktion definiert und aufruft. Ihre Aufgabe ist es, die Lücken so zu füllen, dass sich eine korrekte Funktion und ein korrektes Programm ergibt! Beantworten Sie die Fragen auf der rechten Seite!

```
// POST: writes the zip of [xbegin, xend) and [ybegin, yend) to standard output
void zip (const int* xbegin, const int* xend,
          const int* ybegin, const int* yend)
{
  const int* xp = A ;
  const int* yp = B ;
  while ( C ) {
    std::cout << D << ' ' << E << ' ';
    F ;
    G ;
  }
}

int main ()
{
  int x[] = {1, 3, 5, 7, 9, 11};
  int y[] = {2, 4, 6, 8};
  zip (x, H , y, I ); // 1 2 3 4 5 6 7 8
  return 0;
}
```

Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_m)$ be two sequences of possibly different length. We define the zip of x and y as

$$\mathrm{zip}(x, y) = (x_1, y_1, x_2, y_2, \ldots, x_{\min(n,m)}, y_{\min(n,m)}).$$

For example, $x = (1, 3, 5, 7, 9, 11)$ and $y = (2, 3, 4, 6, 8)$ yields $\mathrm{zip}(x, y) = (1, 2, 3, 4, 5, 6, 7, 8)$.

We are looking for a function that receives two integer sequences x and y as ranges and outputs $\mathrm{zip}(x, y)$. Above, you find a skeleton of a program that defines and calls such a function. Your task is to fill the gaps in such a way that you get a correct function and a correct program. Answer the questions on the right-hand side!

(a) Welcher Ausdruck muss bei A eingesetzt werden? Which expression has to be replaced for A? 0.5 P

A:

(b) Welcher Ausdruck muss bei B eingesetzt werden? Which expression has to be replaced for B? 0.5 P

B:

(c) Welcher Ausdruck muss bei C eingesetzt werden? Which expression has to be replaced for C? 2 P

C:

(d) Welcher Ausdruck muss bei D eingesetzt werden? Which expression has to be replaced for D? 0.5 P

D:

(e) Welcher Ausdruck muss bei E eingesetzt werden? Which expression has to be replaced for E? 0.5 P

E:

(f) Welcher Ausdruck muss bei F eingesetzt werden? Which expression has to be replaced for F? 0.5 P

F:

(g) Welcher Ausdruck muss bei G eingesetzt werden? Which expression has to be replaced for G? 0.5 P

G:

(h) Welcher Ausdruck muss bei H eingesetzt werden? Which expression has to be replaced for H? 0.5 P

H:

(i) Welcher Ausdruck muss bei I eingesetzt werden? Which expression has to be replaced for I? 0.5 P

I:

8 Argumenttypen *(8 Punkte)*

Auf der rechten Seite finden Sie Funktionsdeklarationen, bei denen jeweils die Argumenttypen weggelassen sind. In jedem Fall spezifiziert die Vorbedingung das Verhalten der Funktion. Ihre Aufgabe ist es, die dazu passenden Argumenttypen einzusetzen. Beantworten Sie die Fragen auf der rechten Seite!

On the right-hand side, you find function declarations in which the argument types have been omitted. In each case, the precondition specifies the behavior of the function. Your task is to provide the matching argument types. Answer the questions on the right-hand side!

(a) Ergänzen Sie korrekte Argumenttypen! Provide matching argument types! 2 P

```
// POST: erhoeht den Wert der Variablen balance um amount und gibt
//       den neuen Wert von balance zurueck
//       increases the value of the variable balance by amount and
//       returns the new value of balance
float top_up (                    balance,                    amount);
```

(b) Ergänzen Sie die korrekten Argumenttypen! Provide the matching argument types! 2 P

```
// PRE: [begin, end) ist ein nichtleerer Bereich von double-Werten
//      [begin, end) is a valid nonempty range of double-values
// POST: gibt den Durchschnitt der Werte in [begin, end) zurueck
//       returns the average of all values in [begin, end)
double average (                  begin,                    end);
```

(c) Ergänzen Sie die korrekten Argumenttypen! Provide the matching argument types! 2 P

```
// POST: der Bereich [begin, end) von ganzen Zahlen wird in
//       aufsteigender Reihenfolge sortiert
//       the range [begin end) of integers is sorted in
//       increasing order
void sort (                       begin,                    end);
```

(d) Ergänzen Sie die korrekten Argumenttypen! Provide the matching argument types! 2 P

```
// POST: gibt zurueck, ob die natuerliche Zahl n eine Primzahl
//       ist; falls nein, wird factor auf einen echten Teiler
//       von n gesetzt
//       returns whether the natural number n is prime. If not,
//       factor is set to a proper divisor of n
bool is_prime (                   n,                        factor);
```