

Informatik 1 – Kurzprüfung 2 – LÖSUNG

Herbstsemester 2013
Dr. Felix Friedrich
4.12.2013

Name, Vorname:

Legi-Nummer:

Assistent:

Allgemeine Richtlinien und Informationen:

- Dauer der Prüfung: 25 Minuten.
- Keine Hilfsmittel erlaubt.
- Benutzen Sie Kugelschreiber oder Tinte, keinen Bleistift.
- Für die Bewertung werden nur Antworten in den dafür vorgesehenen Rahmen berücksichtigt.
- Sie können die Aufgaben in beliebiger Reihenfolge antworten.
- Mit untenstehender Unterschrift bestätigen Sie, dass Sie diese Prüfung unter regulären Bedingungen ausführen konnten und dass Sie diese Instruktionen verstanden haben.

General guidelines and information:

- *Exam duration: 25 minutes.*
- *No supplementary material permitted.*
- *Use a pen (not a pencil).*
- *Only solutions in the provided answer frames will be taken into account for the evaluation.*
- *You can freely choose the order in that you want to solve the problems.*
- *With your signature below you confirm that you were able to take this exam under regular conditions and that you have read and understood these instructions.*

Unterschrift:

Aufgabe	1	2	3	4	5	6	Σ
Punkte							
(Maximal Punkte)	(3)	(4)	(3)	(3)	(4)	(3)	(20)

Erreichte Punkte / 4 = Examenspunkte arithmetisch gerundet auf 1/2 – maximal 5

Aufgabe 1. (3 Punkte)

Kreuzen Sie an, ob die Aussagen richtig oder falsch sind.

Mark if the statements are correct or wrong.

Aussage <i>Statement</i>	correct richtig	wrong falsch
Im Gegensatz zu Zeigern gibt es keine Nullreferenzen. <i>In contrast to pointers, there are no null-references.</i>	x	
Der Rückgabetypp einer Funktion darf keine Referenz sein. <i>The return type of a function may not be a reference.</i>		x
Eine nicht const-Referenz kann nicht verwendet werden, um eine const-Variable zu referenzieren. <i>A non-const reference cannot be used to refer to a const variable.</i>	x	
Eine Referenz muss immer mit einem L-Wert initialisiert werden. <i>A reference must always be initialized with an l-value.</i>		x
Der Wert eines Objekts vom Typ <code>const T*</code> kann nicht modifiziert werden <i>The value of an object of type <code>const T*</code> cannot be modified</i>		x
Der Wert eines Objekts vom Typ <code>T* const</code> kann nicht modifiziert werden <i>The value of an object of type <code>T* const</code> cannot be modified</i>	x	

Aufgabe 2. (4 Punkte)

Gegeben seien die folgenden beiden Funktionen f und g.

Consider the following two functions f and g.

```
int f(const int *begin, const int* end) {
    int val1 = *begin;
    if (++begin != end) {
        const int val2 = f(begin, end);
        if (val2 < val1) { val1 = val2; }
    }
    return val1;
}

int g(const int *begin, const int* end) {
    int val= *begin;
    for (const int* it=++begin; it != end; ++it) {
        if (*it > val) { val = *it; }
    }
    return val;
}
```

Kreuzen Sie an, ob die Aussagen wahr oder falsch sind.

Mark if the statements are correct or wrong

Aussage Statement	richtig correct	falsch wrong
Die Funktionen f und g haben dieselbe Vorbedingung Functions f and g have the same precondition.	x	
Die Funktionen f und g haben dieselbe Nachbedingung Functions f and g have the same postcondition.		x

Betrachten Sie nun folgendes Programmstück

Now consider the following program

```
{
    const int my_size = 6;
    int my_values[my_size] = { 10, 5, 7, 9, 20, 4 };
    int result = f(my_values, my_values+my_size);
}
```

Beantworten Sie folgende Fragen

Answer the following questions:

Frage question	Antwort answer
Was ist der Wert der Variablen result nach dem Aufruf von f? What is the value of variable result after call of f?	4
Wie oft wird f insgesamt aufgerufen How many times will function f be called ?	6

Aufgabe 3. (3 Punkte)

Gegeben sei folgender Code

Consider the following code

```
int a = 5;
int* b = &a;
int& c = a;
```

Kreuzen Sie in der folgenden Tabelle an, welche Ausgabe ein Ausdruck bewirkt oder „ungültig“, wenn er ungültig ist.

In the following table, mark what kind of output an expression provides or "invalid" if it is invalid.

Ausdruck Expression	Resultat / result				
	Ungültig invalid	5 5	Adresse von a address of a	Adresse von b address of b	Adresse von c address of c
std::cout << b			x		
std::cout << *b		x			
std::cout << &b				x	
std::cout << c		x			
std::cout << *c	x				
std::cout << &c			x		

Aufgabe 4. (3 Punkte)

Folgender Code bearbeitet den Inhalt eines Arrays *a*. Geben Sie den Inhalt jedes Elements von *a* zum jeweiligen Ausführungszeitpunkt in den dafür vorgesehenen Feldern an.

The following code manipulates the content of the array *a*. Provide the content of each element of *a* at the corresponding point of execution in the respective fields.

```
int a[8] = {2, 0, -2, 10, -3};
```

2	0	-2	10	-3	0	0	0
---	---	----	----	----	---	---	---

```
a[7] = 5;
*(a + 3) = 4;
```

2	0	-2	4	-3	0	0	5
---	---	----	---	----	---	---	---

```
int* p = a + 4;
*(a + a[3] + *(p - 2)) = *(p + 3) - 9;
```

2	0	-4	4	-3	0	0	5
---	---	----	---	----	---	---	---

Aufgabe 5. (4 Punkte)

Das folgende Gebilde nennt sich Pascalsches Dreieck.

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
  ...
```

Die Zahlen an den Kanten des Dreiecks sind alle 1, und jede Zahl innerhalb des Dreiecks ist die Summe der beiden Zahlen direkt oberhalb davon. Die folgende rekursive Funktion gibt die Zahl an der Zeile r und Spalte c (gezählt ab 0) zurück. Zum Beispiel: $pascal(2,0) = 1$, $pascal(2,1) = 2$ and $pascal(3,1) = 3$. Es fehlen zwei Ausdrücke. Setzen Sie die korrekten Ausdrücke ein.

The following pattern is called Pascal's triangle.

The numbers at the edges of the triangle are all 1, and each number inside the triangle is the sum of the two numbers immediately above it. The following recursive function returns the number at row r and column c (counting from 0). For example, $pascal(2,0) = 1$, $pascal(2,1) = 2$ and $pascal(3,1) = 3$. However, two expressions are missing. Find them back!

```
//PRE: Two numbers r and c such that c <= r
//POST: the number at row r and column c
unsigned int pascal(const unsigned int r, const unsigned int c)
{
    if ( $c == 0 || r == c$ )
        return 1;

    return  $pascal(r - 1, c - 1) + pascal(r - 1, c)$ ;
}
```

Aufgabe 6. (3 Punkte)

Schreiben Sie in folgendem Programm die generierte Ausgabe des jeweiligen Funktionsaufrufes in die Rechtecke.

In the following program write the output generated by the respective function calls into the rectangles.

```
#include <iostream>

enum Color {White, Black, Red, Green, Blue};

struct Node {
    int value;
    Node* next;
    Node(int v, Node* n): value(v), next(n) {}
};

void f(const Node* node){
    if (node)
    {
        std::cout << node->value << " ";
        f(node->next);
    }
}

void g(const Node* node){
    if (node)
    {
        g(node->next);
        std::cout << node->value << " ";
    }
}

int main()
{
    Node* first = 0;
    first = new Node (Red, first);
    first = new Node (Blue, first);
    first = new Node (White, first);
    first = new Node (Black, first);

    f(first); 1 0 4 2

    g(first); 2 4 0 1

}
```