

Name, Vorname:

Legi-Nummer:

Diese Selbsteinschätzung dient Ihrer und unserer Orientierung. Sie wird eingesammelt, korrigiert und vertraulich behandelt. Sie hat keinen Einfluss auf eine spätere Leistungsbewertung. **Sie haben 20 Minuten Zeit.**

Das folgende Kleingedruckte finden Sie auch auf einer "scharfen" Prüfung.

Allgemeine Richtlinien:

General guidelines:

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. Dauer der Prüfung: 20 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 nicht berücksichtigt. 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. 10. Richtig beantwortete Multiple-Choice Fragen erhalten jeweils 1 Punkt. Es gibt keine negativen Punkte in MC Fragen. | <p><i>Exam duration: 20 minutes.</i></p> <p><i>Permitted examination aids: dictionary (for spoken languages). No sheets of your own! We will give you extra sheets on demand.</i></p> <p><i>Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.</i></p> <p><i>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 not be considered.</i></p> <p><i>There are no negative points for false answers.</i></p> <p><i>If you feel disturbed by anyone or anything, let the supervisor of the exam know immediately.</i></p> <p><i>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.</i></p> <p><i>If you need to go to the toilet, raise your hand and wait for a supervisor.</i></p> <p><i>We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.</i></p> <p><i>Correctly answered Multiple-Choice questions receive 1 point each. There are no negative points in MC questions</i></p> |
|--|--|

Aufgabe	1	2	3	4	5	6	Σ
Punkte							
Maximum	4	6	6	6	6	8	36

1 Characters (4 Punkte)

Betrachten Sie die folgende Schleife und geben Sie ihre Ausgabe direkt in dem dafür vorgesehenen Kästchen an. Nehmen Sie an, dass Buchstaben gemäss dem ASCII-Code sortiert sind!

Consider the following loop and write its output directly in the provided box. Assume that letters are ordered according to the ASCII code!

```
for (char c = 'F'; c < 'R'; c += 3)
    std::cout << c;
```

Ausgabe *output*:

2 Pointers (6 Punkte)

Gegeben sei folgender Code *Consider the following code*

```
int a = 4;
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 <i>Expression</i>	Resultat / <i>result</i>			
	Ungültig <i>invalid</i>	4	Adresse von a <i>address of a</i>	Adresse von c <i>address of c</i>
std::cout << b				
std::cout << *b				
std::cout << &b				
std::cout << c				
std::cout << *c				
std::cout << &c				

3 Referenzen (6 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
Referenzen können benutzt werden, um bei Funktionsaufrufen unnötige Kopien der Argumente zu verhindern. <i>References can be used to prevent unnecessary copies of the arguments of function calls.</i>		
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>		
Eine Referenz muss immer mit einem L-Wert initialisiert werden. <i>A reference must always be initialized with an L-value.</i>		
Eine Funktion sollte niemals eine Referenz auf eine lokale Variable zurückgeben. <i>A function should never return a reference to a local variable.</i>		
Wenn ein Zeiger des Typs <code>const T*</code> auf ein Objekt im Speicher vom Typ <code>T</code> zeigt, kann dieses Objekt seinen Wert nicht ändern. <i>If a pointer of type <code>const T*</code> points to an object of type <code>T</code> in memory, this object cannot change its value.</i>		
Nachdem eine Referenzvariable initialisiert wurde, kann sie jede andere Variable des gleichen Typs referenzieren. <i>Once a reference variable has been initialized, it can refer to any other variable of the same type.</i>		

4 Iteration (6 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 <i>Statement</i>	richtig correct	falsch wrong
Die Funktionen f und g haben dieselbe Vorbedingung <i>Functions f and g have the same precondition.</i>		
Die Funktionen f und g haben dieselbe Nachbedingung <i>Functions f and g have the same postcondition.</i>		

Betrachten Sie nun folgendes Programmstück

Now consider the following program

```
{  
    const int my_size = 7;  
    int my_values[my_size] = { 3, 8, 1, 12, 20, 2, 6 };  
    int result = f(my_values, my_values+my_size);  
}
```

Beantworten Sie folgende Fragen

Answer the following questions:

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

5 Felder und Arrays I (6 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] = {3, 1, -3, -2, 10};
```

--	--	--	--	--	--	--	--

```
a[7] = 5;
```

```
*(a + 3) = 4;
```

--	--	--	--	--	--	--	--

```
int* p = a + 4;
```

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

--	--	--	--	--	--	--	--

6 Felder und Zeiger II (8 Punkte)

Seien x und y zwei Folgen mit Längen n und m , $x = (x_1, x_2, \dots, x_n)$ und $y = (y_1, y_2, \dots, y_m)$. Wir definieren den Reissverschluss von x und y als

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

Gesucht ist eine Funktion, die zwei Folgen ganzer Zahlen x und y als Zeigerbereiche erhält und $\text{zip}(x, y)$ ausgibt. Hier ist ein Beispiel:

$$x = (1, 3, 5, 7, 9, 11), y = (2, 4, 6, 8) \Rightarrow \text{zip}(x, y) = (1, 2, 3, 4, 5, 6, 7, 8).$$

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!

Let x and y be two sequences of lengths n and m , $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_m)$. We define the zip of x and y as

We are looking for a function that receives two integer sequences x and y as ranges and outputs $\text{zip}(x, y)$. Here is an example:

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!

```
// 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 = ;
    const int* yp = ;
    while (  ) {
        std::cout <<  << ' ' <<  << ' ';
        ++xp;
        ++yp;
    }
}

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