Technical University of Munich
School of Engineering and Design
Prof. Dr. Martin Werner

2nd Exam Computational Foundations I (LRG0060, WS 2021/22)

Rules:

- Keep these sheets closed until the official start of the examination is announced.
- Already now, write your name and matrikel number onto this front page. Have your passport and student ID available on your desk.
- No further items except this examination document, paper given out by us, a few pens, and passport / student ID are allowed on your desk.
- If you have a question, raise your hand. If the answer is relevant to everyone, we will share it with everyone.
- If you are done, please stay in your location quietly and double-check everything. If you need to go to the restrooms, raise your hands. Only one person is allowed to leave the room at the same time.

Last Name: _____

Given Name: _____

Matrikel No.: _____

Evaluation

Task 1: Multiple Choice	/ 5
Task 2: Algorithm Representation	/ 5
Task 3: Program Analysis	/ 15
Task 4: Turing Machine	/ 10
Task 5: Markov Algorithms	/ 10
Task 6: Understanding Recursion	/ 5
Task 7: The Structured Program Theorem	/ 5
Task 8: Composition and Recursive Data Structures	/ 5
Task 9: Faculty	/ 10
Sum	/ 70

Task 1: Multiple Choice

(5 points.)

Answer the following questions with yes or no.
A wrong answer is counted as -1, a correct answer is counted as +1, an answer not given is counted as 0 (if you don't know the answer it is smarter not to give an answer!). If you reach more than 5 points, the points will become bonus points, if you reach less than 0 points, the result is 0 points.

Are the following statements true or false?		True	False
a	In C++, objects can be allocated with `new`. In order to release them, one uses the function `free`.		
b	A Markov algorithm can not be implemented on a Turing machine.		
c	For arbitrary large n, an algorithm of runtime complexity $O(n^2)$ will become faster than an algorithm of $O(n)$.		
d	The SDL libraries allow for visualizing things from C++.		
e	MATLAB is an interpreted language. Hence, MATLAB code runs typically faster than code written in a compiled language.		
f	C++ is an extension to C.		
g	A binary tree with depth n has at most $\sum_{l=0}^{n-1} 2^l$ nodes.		

Task 2: Algorithm Representation

(5 points.)

Consider the Algorithm in Listing 1

Listing 1: An imperative program

```
1    DECLARE A,B,C,D,E,RESULT
2    A = 4;
3    B = SQUARE(A);
4    C = 2;
5    D = 5;
6    E = C+D;
7    RESULT = E*B;
```

a. Write a table with a row for each line of the previous program and a column for each variable declared and give the value **after** the line has completed. Use a ? if the value is unknown (e.g., the variable has not yet been initialized). Include Line 1 (which just means every variable value is unknown).

b. Consider the following program:

```
1 A = CIRCLE(P, R);
2 B = CIRCLE(Q, R);
3 M1, M2 = INTERSECT(A,B);
4 L = LINE(M1,M2)
5 RESULT = INTERSECT(L, LINE(P,Q))
```

Give the same program as a single expression for result. That is, not using multiple lines of imperative code, but braces to organize the computation of the result.

c. Assume P, Q are given points, R is the radius of the circles and the length of the segment from P to Q, INTERSECT returns two intersection points of geometries (either of two circles or of two lines), LINE constructs a line between two points. Draw what the program does. Give variable names as labels for the objects you draw.

Task 3: Program Analysis

(15 points.)

Consider the C++ program in Listing 2. The function print just outputs a space-separated list of the values of the argument vector and its implementation details can be ignored.

Listing 2: Nice Program

```
#include<iostream>
#include<vector>
#include<iterator>
using namespace std;
void print(vector<int> const &input)
{
    std::cout << input[0] << input[1]<< ":" <<input[2]<<input[3] << std::endl;
}

void f(vector<int> &a, int where=3)
{
    a[where] ++;
    switch(where){
       case 1:
       case 3:
         if (a[where] >9)
       a[where] = 0;
         break;
      case 2:
         if (a[where] >5)
       a[where] = 0;
   break;
      case 0:
         if (a[where] >2)
       a[where] = 0;
   break;
    }
    if (a[where] == 0 && where != 0)
       f(a, where-1);
}
int main(void)
{
   vector<int> A {0,8,1,5};
   print(A);f(A);
   print(A);f(A);
   while ( A[0] != 1 || A[1] != 2 || A[2] != 0 || A[3] != 0)
   {
     f(A);
   }
   print(A);
}
```

a. How often is the loop executed?

b. Give the complete output of the program.

c. Define the term “Loop Variant” in one sentence. Only the first given sentence is taken into account.

Task 4: Turing Machine

(10 points.)

Given a Turing machine with a finite string of 0s and 1s on the tape, delimited in both directions with empties ϵ, implement an algorithm that interprets the tape as an unsigned integer numbers and computes the increment. Implement the Turing machine with no more than three states, where the final state is called `done` and has no outgoing rules.

a. Give the machine as a transition table. The table can be incomplete. As agreed, the machine will halt if there is no applicable rule.

b. Which of the following properties do Turing machines have? Answer yes or no:

- they are determined:
- they are deterministic:
- they are capable of computing Markov Algorithms:

Task 5: Markov Algorithms

(10 points.)

Markov algorithms have been introduced to simplify the analysis and implementation of Semi Thue Systems.

Listing 3: A Markov Algorithm

```
1 |0 -> 0||
2 1 -> 0|
3 0 -> ϵ
```

a. What is the key difference (1 sentence) between Semi Thue systems and Markov algorithms?

b. Execute the given Markov algorithm on the input strings “101” and “11”. Give all intermediate strings and (preferably for correction) the line number of the rule you applied. Tip: Write your transitions below each other as most rules don’t change the length. Mark the left hand side (e.g., underline), replace it, and fill in the rest.

c. In general,what does this program do? (One Sentence)

Task 6: Understanding Recursion

(5 points.)

Consider the following program

Listing 4: "Recursion"

```
1  #include<iostream>
2  int B(int v);
3
4  int A(int v)
5  {
6     if (v >4) return B(v+2);
7     return A(v*3);
8  }
9  int B(int v)
10 {
11    while (v>4) return B(v-4);
12    return v-1;
13 }
14
15 int main()
16 {
17    std::cout << "A(" << 4 << ") == " << A(4) << std::endl;
18    std::cout << "A(" << 5 << ") == " << A(5) << std::endl;
19 }
```

a. First, give all function calls with parameters (written like "$A(4) \Rightarrow A(5) \Rightarrow B(2)$") and the final value. Then, give the exact output of the program.

b. [2 Bonus Points] For which inputs does the function A terminate? Give a short reason. Ignore possible overflow or underflows from finite arithmetic representation.

Task 7: The Structured Program Theorem

(5 points.)

We introduced the Turing machine as a basic model for computing. However, programming of computers is not done in the language of Turing machines in practice. The Structured Program Theorem names three concepts that a programming language must fulfill to efficiently express all Turing programs.
Name these three concepts and draw them as a flow chart.

Task 8: Composition and Recursive Data Structures

(5 points.)

In the lecture, we have learnt about recursive data structures. Declare a class in C++ which could be used to implement a binary tree. Give no methods except a constructor that ensures proper initialization of the needed pointer variable. (The answer will typically have less than 10 lines, we are not picky about syntax. If it is semantically sensible, you get all points!)

Task 9: Faculty

(10 points.)

Sketch two implementation (we don't care about syntactic details) of a function that calculates the sum

$$\sum_{i=0}^{N} 2 \cdot i$$

The first version shall be a recursive one and the second version shall be an imperative one. Both function should follow the signare
`int specialsum(int N) {...}`.