ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE – LAUSANNE
POLITECNICO FEDERALE – LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY – LAUSANNE

Faculté Informatique et Communication
Cours d'Informatique à la section SV
B. Jobstmann, J. Sam

ICC

Examen Semestre I

Instructions :

— You have 3 hours to complete this exam (8h00 - 11h).

— The maximum number of points is 100.

— **Note that there are also instructions on the back side.**

— You must **write using black or dark-blue ink**, do not use pencils or other colors. Do note use **erasable pens** (loss of information due to heat).

— Any paper document is permitted. However, you are not allowed to use a personal computer, nor a cellphone, nor any other electronic device.

— Reply on the answering sheets that were distributed to you **in the dedicated places**. **Do not answer on the statement**.

— Do not attach any additional piece of paper ; **only the distributed answering sheets will be graded**.

— You can reply to the questions in English or in French.

— The exam is made of 5 independent exercises (2 for theory and 3 for programming).

You can start with whatever exercise you want

Exercise	1	2	3	4	5
Points	20	16	15	34 (13 + 21)	15

suite au verso ☞

Exercise 1 : MC Theory part [20 points]

For each multiple-choice question, you can get p points. For partially correct answers, you will get partial points according to the formula "$p \cdot \max(0, x_C/C - x_I/I)$", where C and I are the total numbers of correct and incorrect options, respectively, and x_C and x_I are the numbers of correct and incorrect options that you selected. **The total number of points you receive per question is never negative.**

QCM 1.1 : Landau Notations [2 points]

Consider the following statements that use different Landau notations. Which statements are correct ? (**Multiple correct answers are possible.**)

A) $3 \cdot n + n^2 + 100 \in \Omega(n)$
B) $n^{10} + 2^n \in O(n^{10})$
C) $n \cdot \log n + 4 \cdot n^3 + n \in \Theta(n^3)$
D) $n^3 + n + 100 \in \Theta(n^4)$

Algorithms and their Complexity

Consider the following algorithms ALGO1 and ALGO2. ALGO1 takes two numbers as input and returns a number. The second algorithm ALGO2 takes two lists L_1 and L_2 of numbers as input and returns a list of numbers. We use the notation $\{\}$ to refer to an empty list and APPEND(L, x) to append the element x to the end of the list L. We use the notation $L[i]$ to access the i-th element in the list L. We start counting at 1. So, the list L has the elements $L[1], L[2], \ldots L[n]$, where n is the number of elements in the list. In addition, the algorithm SIZE(L) takes the list L as an input and returns the number of elements in L.

Algorithm 1 ALGO1(a, b)

```
1 : $m \leftarrow a$
2 : if $b < m$ then
3 :     $m \leftarrow b$
4 : return $m$
```

Algorithm 2 ALGO2(L_1, L_2)

```
1 : $n_1 \leftarrow$ SIZE$(L_1)$
2 : $n_2 \leftarrow$ SIZE$(L_2)$
3 : $L \leftarrow \{\}$
4 : $m \leftarrow$ ALGO1$(n_1, n_2)$
5 : for $i$ from 1 to $m$ do
6 :     APPEND$(L, L_1[i] + L_2[i])$
7 : for $i$ from $m + 1$ to $n_1$ do
8 :     APPEND$(L, L_1[i])$
9 : for $i$ from $m + 1$ to $n_2$ do
10 :     APPEND$(L, L_2[i])$
11 : return $L$
```

QCM 1.2 : Execution of an Algorithm [2 points]

Which of the following lists is returned when **Algorithm 2** is called with the lists $\{1, 2, 4, 5\}$ and $\{3, 2, 1, 5, 1, 2\}$, i.e., ALGO2($\{1, 2, 4, 5\}$, $\{3, 2, 1, 5, 1, 2\}$).

A) $\{1, 2, 4, 5, 3, 2, 1, 5, 1, 2\}$
B) $\{1, 3, 2, 2, 4, 1, 5, 5, 1, 2\}$
C) $\{4, 4, 5, 10, 1, 2\}$
D) $\{3, 2, 2, 7, 5, 7\}$

QCM 1.3 : Complexity [2 points]

Let m be the maximum of the lengths of the two lists L_1 and L_2, i.e., $m = \max(\text{size}(L_1), \text{size}(L_2))$, and assume that $T(m)$ describes the number of instructions that ALGO2(L_1, L_2) executes in the worst case depending on m. We assume that the number of instructions to create an empty list ($\{\}$), to add an element to a list (APPEND), and to compute the size of a list (SIZE) is constant. Which statement is correct ?

A) $T(m) \in \Theta(\log m)$
B) $T(m) \in \Theta(m)$
C) $T(m) \in \Theta(m^2)$
D) $T(m) \in \Theta(m^3)$

Recursive Algorithms and their Complexity

Consider **Algorithm 3**, which takes two numbers n and b as inputs and returns a list of numbers. We use the same notations for lists ($\{\}$, APPEND, and SIZE) as for **Algorithm 2**. The slash (/) refers to the integer division, i.e., the division in which the fractional part is discarded, e.g., $5/2 = 2$.

Algorithm 3 CREATELIST(n, b)

```
1 : $L \leftarrow \{\}$
2 : if $n <= 1$ then
3 :     APPEND$(L, b + n)$
4 : else
5 :     $L \leftarrow$ CREATELIST$(n/2, b)$
6 :     for $i$ from 1 to $n$ do
7 :         APPEND$(L, b + n)$
8 : return $L$
```

QCM 1.4 : Recursive Algorithm [2 points]

Which of the following lists is returned when **Algorithm 3** is called with the inputs $n = 4$ and $b = 3$, i.e., CREATELIST$(4, 3)$.

A) $\{4, 5, 6, 7\}$
B) $\{4, 5, 5, 7, 7, 7, 7\}$
C) $\{4, 5, 5, 6, 6, 6, 7, 7, 7\}$
D) $\{4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7\}$

QCM 1.5 : Complexity [2 points]

What is the (asymptotic worst-case) complexity of **Algorithm 3** if we assume that appending an element to a list is in $\Theta(1)$?

A) $\Theta(\log n)$
B) $\Theta(n)$
C) $\Theta(n \cdot \log n)$
D) $\Theta(n^2)$

QCM 1.6 : Computability [2 points]

Which of the following statements are correct ? (**Multiple correct answers are possible.**)

A) We can find an algorithm that gives the correct answer for all instances of the halting problem.
B) The problem of sorting a list is in the complexity class NP.
C) We do not know if the complexity class P is equal to the complexity class NP.
D) If a problem is in the complexity class NP, then we can find a polynomial-time algorithm that gives the correct answer for all instances of the problem.

QCM 1.7 : Computation Architecture [2 points]

Which of the following statements are correct ? (**Multiple correct answers are possible.**)

A) A computer that implements the von Neumann architecture can execute an arbitrary sequence of instructions taken from the list of instructions this computer accepts.
B) All CPUs in the world use the same instruction set.
C) Storage elements, e.g., the instruction pointer, can be implemented with transistors.
D) All of the performance improvements of CPUs over the last 30 years were due to smaller and faster transistors.

QCM 1.8 : Circuits [2 points]

Recall that we use the symbols shown in Figure 1 to represent AND, OR, and NOT gates, respectively.

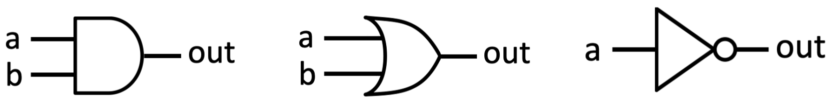

FIGURE 1 – Symbols for an AND gate (left), an OR gate (middle), and a NOT gate (right).

Consider the circuit given in Figure 2. Which of the following (incomplete) truth tables describes the functionality of this circuit ?

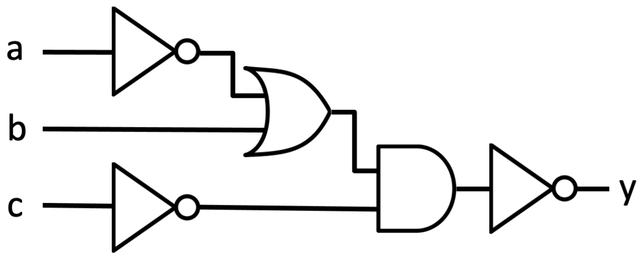

FIGURE 2 – Circuit

Table A)

a	b	c	y
0	0	0	0
..	..	..	..
1	1	0	0
1	1	1	1

Table B)

a	b	c	y
0	0	0	0
..	..	..	..
1	1	0	1
1	1	1	1

Table C)

a	b	c	y
0	0	0	1
..	..	..	..
1	1	0	0
1	1	1	1

Table D)

a	b	c	y
0	0	0	0
..	..	..	..
1	1	0	0
1	1	1	0

QCM 1.9 : Assembly Language [2 points]

Assume you are given a CPU with ten registers (`r0` to `r9`) and the basic instructions given in Table 1.

Table 1 – Instruction Set

Instruction	Meaning
`copy r0, c`	$r0 \leftarrow c$: copy a constant c into register r0
`copy r0, r1`	$r0 \leftarrow r1$: copy the value of register r1 to register r0
`add r0, r1, c`	$r0 \leftarrow r1 + c$: add constant c to the register r1 and put the result in r0
`add r0, r1, r2`	$r0 \leftarrow r1 + r2$: add the values of registers r1 and r2 and put the result in r0
`sub r0, r1, c`	$r0 \leftarrow r1 - c$: subtract the constant c from the register r1 and put the result in r0
`sub r0, r1, r2`	$r0 \leftarrow r1 - r2$: subtract the value of register r2 from r1 and put the result in r0
`mul r0, r1, r2`	$r0 \leftarrow r1 \cdot r2$: multiply the values of register r1 and r2 and put the result in r0
`div r0, r1, r2`	$r0 \leftarrow r1/r2$: integer division of register r1 by r2 ; the result is stored in r0
`jump n`	Jump to line n of the program
`jump_ge r0, r1, n`	Jump to line n if the value in r0 is greater than or equal to the value in r1 ($r0 \geq r1$)
`jump_ge r0, c, n`	Jump to line n if the value in r0 is greater than or equal to the constant c ($r0 \geq c$)
`stop`	Stop the program

Consider the assembly program shown on the right. What is the value of `r2` after executing this assembly program, if `r0` is initially 2 and `r1` is initially 3 ?

A) 2

B) 5

C) 9

D) 14

```
1: copy r2, 0
2: copy r3, 0
3: copy r4, 0
4: jump_ge r3, r1, 9
5: add r4, r0, r3
6: add r2, r2, r4
7: add r3, r3, 1
8: jump 4
9: stop
```

QCM 1.10 : Memory [2 points]

Suppose that the main memory (RAM) of a computer has 16 blocks of 2 words each (32 words in total), and that each block is indexed by the address of its first word : $0, 2, 4, 6, 8, 10, \ldots, 30$. In addition, suppose that this computer has a cache that can hold up to 3 blocks and that the blocks 4 and 20 are already loaded into the cache. How many cache misses occur when the processor uses the LRU (Least Recently Used) algorithm and accesses the following sequence of memory addresses ?

$$5 \quad 10 \quad 21 \quad 24 \quad 4$$

A) 2
B) 3
C) 4
D) 5

Exercise 2 : Writing an algorithm [16 points]

A) Write an algorithm called sumRow that given a matrix M of non-negative number and a row index r, returns the sum of all elements in the row r. The matrix is given as a list of lists, e.g.,

$$M = \{\{1, 2, 4, 6\}, \{0, 1, 3, 1\}, \{1, 1, 1, 1\}\}$$

refers to a matrix with three rows and four columns. If M is the matrix above and r is 2 then the algorithm should return the sum of the second row $\{0, 1, 3, 1\}$, which is $0 + 1 + 3 + 1 = 5$, i.e., sumRow($\{\{1, 2, 4, 6\}, \{0, 1, 3, 1\}, \{1, 1, 1, 1\}\}, 2$) should return 5. Recall that we use the notation $L[i]$ to access the i element in a list, e.g., $M[1]$ refers to the first row in M, which is $\{1, 2, 4, 6\}$ and $M[1][2]$ refers to the second element in the row $M[1]$, which is 2. In addition, recall that the algorithm size returns the number of elements in a list, e.g., size(M) is 3 and size($M[1]$) is 4. The sum of an empty list is 0. We assume that the given row index r is valid, i.e., $1 \leq r \leq$ size(M).

B) Using the algorithm sumRow (from A) write an algorithm maxRowIndex that finds the index of the row with the largest sum, e.g., maxRowIndex($\{\{1, 2, 4, 6\}, \{0, 1, 3, 1\}, \{1, 1, 1, 1\}\}$) should return index 1. If there are multiple rows with the same maximal sum, the algorithm should return any of these indices. Furthermore, if the matrix is empty the algorithm should return -1.

C) Write an algorithm maxDiff that given a non-empty matrix M of non-negative numbers returns the maximal difference between the sums of any two rows, i.e.,

$$d_{\max} = \max_{i,j \in [1, \textsc{size}(M)]} \left| \sum_{k \in [1, \textsc{size}(M[i])} M[i][k] - \sum_{k \in [1, \textsc{size}(M[j])} M[j][k] \right|.$$

You can use the algorithms written in (A) and (B) and write additional sub-algorithms to simplify your task.

Exercise 3 : C++ Syntax and basics [15 points]

Consider the two following declarations made before the `main` (assume that all necessary libraries have been included) :

```
struct  Trip {
  string destination;
  size_t duration;
};

vector<Trip> trips(5);
```

1. For each of the following statements (taken separately), indicate whether it corresponds to a correct or incorrect code in C++ when used in the `main`. Briefly justify the incorrect cases and specify if the error occurs at compilation time or at runtime.

a) `trips my_trips;`

b) `Trip.destination = "Karakorum";`

c) `Trip  my_trip({1, "Karakorum"});`

d) `trips[10] = {"Karakorum", 1};`

e) `Trip* my_trip(new Trip({"Karakorum", 1}));`

f) `Trip&  my_trip(new Trip({"Karakorum", 1}));`

g) `for (const auto& my_trip : trips) { cout << my_trip.destination << endl; }`

h) `for (const auto& my_trip : trips) { cout << my_trip << endl; }`

i) `for (const auto& my_trip : trips) { my_trip.destination ="Karakorum"; };`

2. For each of the following function prototypes (taken separately), indicate whether it is possible to add it after the two initial declarations. Briefly justify each of your answers.

a) `void f(const Trip& trip);`

b) `void f (Trip trip = {"Karakorum", 1});`

c) `Trip* f();`

d) `void f(*Trip t);`

e) `void f(Trip& t);`

f) `void f(Trip*& t);`

g) `Trip f(&Trip t);`

h) `void f(vector<Trip> trips);`

i) `void f(vector<trips> trips);`

Exercise 4 : Program design and programming [34 points]

A travel agency is seeking your assistance in managing their offer catalog.

A *tourist agency* offers catalogs of travel packages. An *offer* is characterized by a destination (a `string`), its price in the low season (a `double`), the surcharge to apply to this price in the high season, and information indicating whether the destination requires a long-distance flight or not. It is assumed that the destination is unique. A *catalog* is a set of offers. The agency wishes to record : its *catalogs stored by year*[1], the set of its *sales in the low season*, and the set of its *sales in the high season*. It will be considered that these pieces of information characterize the travel agency. A sale is a *pair* that associates a *date* with a travel offer. A date is characterized by a day, a month, and a year. A sale will be considered as made in the high season if its month is between June and September.

Question 1 : Data structures/types [13 points]

Write a C++ code for the data structures/types to model : a date, a offer, a catalog of offers, a sale, and a travel agency.

The definitions of the data structures and types need to be given in the order of the precedence required by C++. You can of course define other types if necessary and make wise use of `typedef`.

Question 2 : Functions [21 points]

The following functionalities are required for our agency. Provide the prototypes for these functions. You are not asked to write a complete program or the function bodies, but only the prototypes. It is also not required to write inclusion directives (`#include<..>`).

You should add other functions if you find them relevant for **good modularization**. You can also **indicate in comments if a function uses others** among those requested if it is necessary for understanding. Please also consider the edge cases.

1. create a travel offer based on its destination and all other information that characterizes it ;
2. calculate the price of an offer depending on whether it would be purchased in high or low season ;
3. calculate the number of sales made either in the low or high season ;
4. find, without displaying, the least sold destination for a given year ;
5. add a new sale (which will be placed in the correct list according to its month) ;
6. find, without displaying, an ecological offer for a given destination in the catalog of a given year ; an offer is ecological if it does not require a long-distance flight ;
7. calculate the amount of ecological sales (sum of all sales regardless of the season) ;
8. find the most sold ecological destination in the low season and the most sold in the high season.

1. For example, the catalog of the year 2023, that of 2022, etc.

Exercise 5 : Program flow [15 points]

The following program compiles and runs without errors (C++11).

```cpp
#include <iostream>
#include <vector>
#include <array>
#include <string>
using namespace std;

struct N {
  vector <int> o;
  vector <int> p;
};
typedef array<int, 3> Row;
typedef array<Row, 3> Set;

typedef string Str;

int c(const Str& str, char car){
  int r(0);
  for (auto c : str) {
    if (c == car) ++r;
  }
  return r;
}
void p (const Row& r) {
  for (auto e : r ) cout << e << " ";
}

void p (const Set& s){
  for (auto e : s ) {
    p(e);
    cout << endl;
  }
}

void p(const N& n){
  for (auto e : n.o) cout << e << " ";
  cout << endl;
  for (auto e : n.p) cout << e << " ";
  cout << endl;
}

int a(const Row& row, int i=0){
  if (i == row.size()) return 0;
  return row[i] + a(row, i+1);
}



```

```

Set i(const string& st){
  Set s;
  size_t w(s.size());
  size_t h(s[0].size());
   for (size_t i(0); i < w; ++i){
    for (size_t j(0); j < h; ++j){
      s[i][j] = c(st, st[i * h + j]);
    }
  }
   return s;
}

N g(const Set& s){
  N n;
  for (auto r : s){
    int val(a(r));
    if (val%2 == 0) n.p.push_back(val);
    else  n.o.push_back(val);
  }
  return n;
}

int main() {
  Str st("AAGGGTTAGCCC");
  Set s(i(st));
  cout << "" << endl;
  p(s);
  cout << "" << endl;
  p(g(s));

  Set* pt(&s);
  cout << "" << endl;
  p(g(*pt));

  return 0;
}
```

1. What does the function c do conceptually ?
2. What does the function a do conceptually ?
3. What does the function g do conceptually ?
4. What does the program display ? Briefly explain its execution. This is not about paraphrasing the code, but rather *explaining* the steps and the execution of the program.