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
Jobstmann B, Sam J.

ICC

Examen Semestre I

Instructions :

- You have 3 hours to complete this exam (8h00 - 11h).
- The maximum number of points is 90.
- 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 6 independent exercises (3 for theory and 3 for programming).

  You can start with whatever exercise you want

Exercise	1	2	3	4	5	6
Points	17	8	15	6.5	30 (7 + 12 + 6 + 5)	13.5

Exercise 1 : MC Theory part [17 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 number of correct and incorrect options that you selected. **The total number of points you receive per question is never negative.**

QCM 1.1 : Circuits [2 points]

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

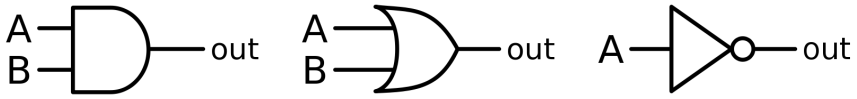

FIGURE 1 – Symbols for an AND gate (left), an OR gate (middle), and a NOT gate (right).

Consider the circuit in Figure 2 and **select all correct statements.**

A) When $D = 1$ the output X of the circuit is always 1.
B) When $A = 0$ the output X of the circuit is always 0.
C) When $C = 1$ and $D = 0$ the output X of the circuit is equal to A.
D) When $B = 1$ and $D = 0$ the output X of the circuit is equal to C.

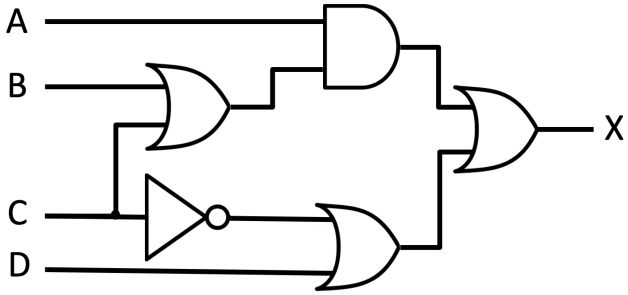

FIGURE 2 – Circuit

QCM 1.2 : Assembly Language [3 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
`multiply r0, r1, r2`	$r0 \leftarrow r1 \cdot r2$: multiply the values of register r1 and r2 and put the result in r0
`divide 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_gt r0, r1, n`	Jump to line n if the value in r0 is greater than the value in r1 ($r0 > r1$)
`stop`	Stop the program

Consider the assembly program shown on the right. What is the value of `r6` after executing this assembly program, if `r0` is initially 3, `r1` is initially 2, and `r2` is initially 10 ?

A) 18

B) 19

C) 20

D) 21

```
 1: copy r6, 1
 2: copy r3, 1
 3: jump_gt r3, r0, 12
 4: copy r4, 1
 5: jump_gt r4, r1, 9
 6: copy r5, r4
 7: add r4, r4, 1
 8: jump 5
 9: multiply r6, r6, r5
10: add r3, r3, 1
11: jump 3
12: add r6, r6, r2
```

QCM 1.3 : Sampling and reconstruction [2 points]

Assume that we would like to sample the signal $X(t) = 5 \cdot \sin(100\pi t + \pi) + 10 \cdot \sin(12\pi t - \pi/2) + 3 \cdot \sin(50\pi t)$. Which of the following sampling frequencies f_e is the smallest frequency that allows a correct reconstruction of the signal $X(t)$ using the interpolation formula $X_I(t) = \sum_n X(nT_e)\text{sinc}(\frac{t-nT_e}{T_e})$ with $T_e = 1/f_e$.

A) $f_e = 50$ Hz **B)** $f_e = 100$ Hz **C)** $f_e = 200$ Hz **D)** $f_e = 300$ Hz

QCM 1.4 : Filters [2 points]

Assume we are using the signal $X(t) = 2 \cdot \sin(8\pi t) + 4 \cdot \sin(12\pi t)$ as input to an ideal low-pass filter with cutoff frequency of f_{c1} and we use $\hat{X}_1(t)$ to denote the output signal of this ideal low-pass filter. In addition assume that $X(t)$ is also used as input to a moving average filter with cutoff frequency of f_{c2}. The output signal of this moving average filter is denoted by $\hat{X}_2(t)$. **Select all correct statements.**

A) If $f_{c1} > 6$, then $X(t) = \hat{X}_1(t)$.

B) If $f_{c1} = f_{c2}$, then $\hat{X}_1(t) = \hat{X}_2(t)$.

C) If $f_{c1} > f_{c2}$, then the bandwidth of $\hat{X}_1(t)$ is always larger or equal to $\hat{X}_2(t)$.

D) If $f_{c1} < f_{c2}$, then the bandwidth of $\hat{X}_1(t)$ is always smaller or equal to $\hat{X}_2(t)$.

QCM 1.5 : Stroboscopic effect [3 points]

Assume that we are sampling the signal $X(t) = \sin(2\pi t) + \sin(4\pi t) + \sin(8\pi t)$ with the sampling frequency $f_e = 6$ Hz and try to reconstruct the signal $X(t)$ using the interpolation formula $X_I(t) = \sum_n X(nT_e)\text{sinc}(\frac{t-nT_e}{T_e})$ with $T_e = 1/f_e$. Select the frequencies that appear in the reconstructed signal $X_I(t)$. (Select one of the options below.)

A) 1 Hz and 2 Hz

B) 1 Hz, 2 Hz, and 4 Hz

C) 2 Hz, 4 Hz, and 6 Hz

D) 2 Hz, 4 Hz, and 8 Hz

QCM 1.6 : Entropy [3 points]

Consider the following four words and order them with respect to their entropy starting with the word with the smallest entropy to the word with the largest entropy.

$w_1 =$ `abbreviation`
$w_2 =$ `facelessness`
$w_3 =$ `obfuscations`
$w_4 =$ `bacchanalian`

Which of the following orderings is correct ?

A) $H(w_2) < H(w_4) < H(w_1) < H(w_3)$

B) $H(w_4) < H(w_2) < H(w_1) < H(w_3)$

C) $H(w_2) < H(w_4) < H(w_3) < H(w_1)$

D) $H(w_4) < H(w_2) < H(w_3) < H(w_1)$

QCM 1.7 : Spectrum [2 points]

Figure 3 shows the frequency spectrum (the red lines) of one of the following signals. Select the signal $X(t)$ from which this spectrum is derived.

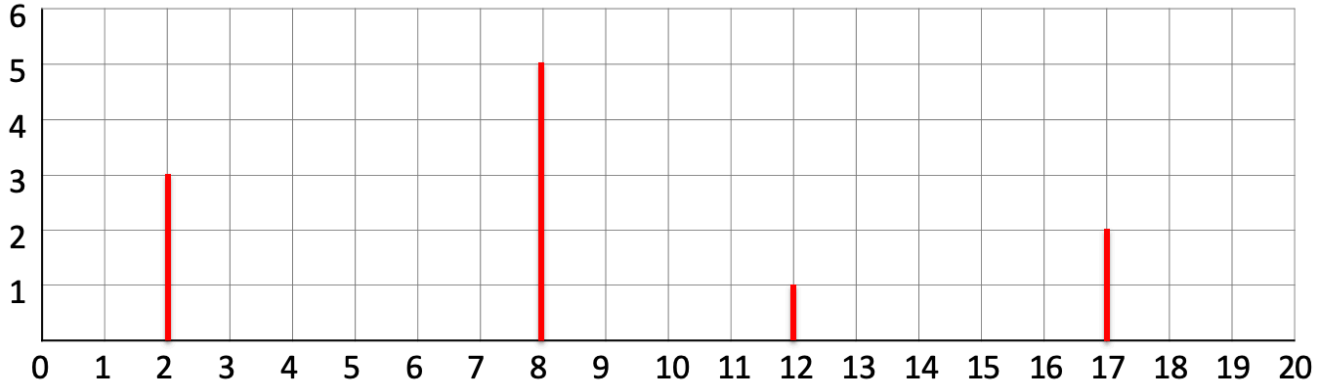

FIGURE 3 – Spectrum

A) $X(t) = 3 \cdot \sin(2\pi t) + 5 \cdot \sin(8\pi t) + \sin(12\pi t) + 2 \cdot \sin(17\pi t)$

B) $X(t) = 6 \cdot \sin(2\pi t) + 10 \cdot \sin(8\pi t) + 2 \cdot \sin(12\pi t) + 4 \cdot \sin(17\pi t)$

C) $X(t) = 3 \cdot \sin(4\pi t) + 5 \cdot \sin(16\pi t) + \sin(24\pi t) + 2 \cdot \sin(34\pi t)$

D) $X(t) = 6 \cdot \sin(4\pi t) + 10 \cdot \sin(16\pi t) + 2 \cdot \sin(24\pi t) + 4 \cdot \sin(34\pi t)$

Exercise 2 : Moving Average filter [8 points]

Draw (on the answering sheet) the output signal $\hat{X}(t)$ of a moving average filter with cutoff period $T = 1$s that takes the periodic signal $X(t)$ given in Figure 4 as input signal. The period of the input signal $X(t)$ is 4s.

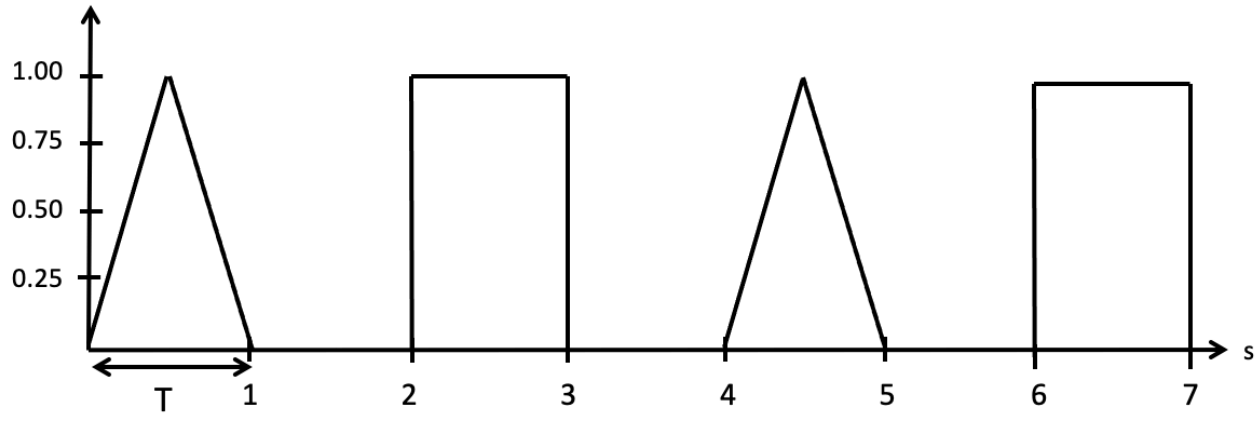

FIGURE 4 – Input signal $X(t)$ of a moving average filter with cutoff period T.

Exercise 3 : Huffman encoding [15 points]

Suppose you would like to send the following sentence (without spaces) to your friend using the minimal number of bits.

ENGAGE LE JEU QUE JE LE GAGNE

In order to create an optimal encoding using Huffman's algorithm and find the minimal number of bits proceed as follows :

3.1 Create a table with the number of appearances per letter (see the table on the answering sheet.)

3.2 Draw a Huffman tree for this table.

3.3 Create a table with a code word for each letter (see the table on the answering sheet.)

3.4 Compute the number of bits needed to send this message with your encoding.

3.5 How many bits do you save with this encoding compare to an ASCII encoding that uses 8 bit per character ?

Exercise 4 : C++ Syntax [6.5 points]

For each of the instructions below, indicate whether it corresponds to a correct or incorrect instruction in C ++. In the case of a correct instruction, briefly explain what the instruction corresponds to (initialization, declaration etc.) :

1. `char c("n");`
2. `vector <vector <int>> v(0,"init");`
3. `f(2) == 3;`
4. `3 = f(2);`
5. `void f = 3;`
6. `f((f(2) == 3));`
7. `constexpr int v(2);`
8. `f(double 3);`
9. `double f(3);`
10. `double f(double);`

Exercise 5 : Program design and programming [30 points]

You are hired to manage a fingerprint database.

A *fingerprint* is characterized by the name of the person to whom it belongs (string of characters) and a black and white image of *pixels*. A pixel can be represented by a Boolean value (e.g. `true` for black and `false` for white). Fingerprint images are fixed size arrays of 100x100 pixels. We suppose that it is possible to calculate the similarity of two fingerprints with an algorithm, whose details do not interest us. This similarity measurement is returned as a positive integer, the larger it is and the similar the fingerprints are. Two fingerprints are similar with a given threshold, `threshold`, if the measure of their similarity is greater than or equal to `threshold`.

A *fingerprint database* is characterized by the year it was created, an identifier (string of characters) and a set of fingerprints.

Question 1 : Data structures/types [7 points]

Provide C++ code for the data types/structures allowing to model : a pixel, a fingerprint image, a fingerprint and a fingerprint database.

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 [12 points]

Below we give the features that are required to manage our fingerprint databases. Give the prototypes of these functions. You are not asked to write a complete program nor the body of functions but only prototypes. You can add other functions if you consider it relevant for a good modularization. You are not asked to write include directives (`#include<..>`).

1. display a fingerprint (its name and associated image) ;
2. display a fingerprint database (its identifier, year and all its fingerprints) ;
3. calculate the similarity of two given fingerprints ;
4. check if two given fingerprints are similar according to a given comparison threshold ;
5. create an empty database (without fingerprints) by taking its identifier and year of creation as parameters ;
6. add a fingerprint, read from a file with a given name, to a given database (we suppose that the file contains the name of the person to whom the fingerprint belongs as well as the Boolean values constituting the image) ;
7. find (without displaying it), in a given database, the fingerprint `e1` which is most similar to a given fingerprint `e2` and return the degree of similarity between `e1` and `e2`.

Question 3 : Data structures and functions (continued) [6 points]

Suppose now that we are interested in coding the following two features (we assume that in these two features we do not compare a fingerprint with itself) :

8. Find (without displaying) all pairs of similar fingerprints according to a given threshold, in a given database ;
9. Given a list of fingerprints, find for each of these fingerprints (without displaying) which other fingerprint in the list is similar according to a given threshold. **For a given fingerprint, there might not exist a sufficiently similar fingerprint**. For example, given the list `{e1, e2, e3, e4, e5}`, we want to obtain a list such that `{{e1, e4}, {e2, e5}, {e3, x}, {e4, e1}, {e5, e2}}`, meaning that `e1` is similar to `e4`, `e2` is similar to `e5` etc. Note that `x` needs to be replaced by a suitable value that indicates that no sufficiently similar fingerprint is found.

What additional (single) type do you propose to represent the notion of *pair* of fingerprints which satisfies the needs of both of the above two functions ?

Propose prototypes of the two additional functions expressed using this new type.

Question 4 : Programming [5 points]

Give the code implementing function 8.

Recall that we do not compare a fingerprint with itself. You can assume that the name of the person who owns the fingerprint uniquely identifies it. Use the prototyped functions without writing their bodies. Be sure to maintain a good indentation.

Exercise 6 : Program flow [13.5 points]

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

```
1  #include <vector>
2  #include <iostream>
3  using namespace std;
4
5  typedef vector <int>  L;
6  typedef vector <L> M;
7  constexpr int MIN_VALUE(-1000);
8
9  void d(ostream& out, const L& lst) {
10     for (const auto& v : lst){
11       out << v << " "<< flush;
12     }
13     out << endl;
14 }
15
16 void d(ostream& out, const M& m){
17   for (const auto& ls : m){
18     d(out,ls);
19   }
20 }
21
22 double m(const L& lst){
23   double v(MIN_VALUE);
24   for (auto i : lst){ if (i > v) v=i; }
25   return v;
26 }
27
28 void  m(const L& lst, int& idx){
29   idx=-1;
30   for (size_t i(0); i < lst.size(); ++i) {
31     if (lst[i] == m(lst)) { idx = i; return; }
32   }
33 }
34
35 int r(L& lst) {
36   int i;
37   m(lst, i);
38   if (i>=0){
39     lst[i] = lst[lst.size()-1];
40     lst.pop_back();
41   }
42   return i;
43 }
44 //SUITE DU CODE SUR LA PAGE SUIVANTE
45
46
47
```

```
48
49 L r(M& m){
50   L res;
51   for (auto& l : m){ res.push_back(r(l));  }
52   return res;
53 }
54
55 M ra(M& m){
56   M res;
57   while (m.size() != m[0].size()){ L sub(r(m)); res.push_back(sub); }
58   return res;
59 }
60
61 int main(){
62   M m3({{ 1, 1,  5,  1,  2, 1},
63         { 2, 1, -4,  1,  1, 1},
64         { 1, 8,  2,  1,  1, 1},
65         { 2, 1, 19,  1,  1, 1}});
66
67   M m4(m3); int k(0);
68
69   cout << ++k << "-------" << endl;
70   d(cout, m3);
71   cout << ++k << "-------" << endl;
72   r(m3);
73   d(cout, m3);
74   cout << ++k << "-------" << endl;
75   d(cout, m4);
76   cout << ++k << "-------" << endl;
77   d(cout, ra(m4));
78   cout << ++k << "-------" << endl;
79   d(cout, m4);
80   return 0;
81 }
```

1. What do the `d` functions do conceptually ?
2. What do the `m` functions do conceptually ?
3. (bonus) What criticism can you make on the efficiency of the function implemented at line 28 ? briefly describe in French how it is possible to improve this ?
4. What do the `r` functions do conceptually ?
5. On line 51, can we delete the symbol `&` while preserving the functionality initially implemented ? Justify briefly.
6. What does the `ra` function do conceptually ?
7. What does the program display ? Briefly explain how it works. Do not paraphrase the code, but *explain* the steps and program flow.