Exercise 1: Language discovery

For each of the following program or fragment of program, please indicate: (a) What the fragment does (b) Is it written using the imperative or functional paradigm (c) In which language the fragment is written.

1. PROGRAM HELLO
   WRITE(6,*), 'HELLO WORLD'
   STOP
   END

2. PROGRAM FACT
   J=1
   DO 1 I=1,10
   J=J*I
   1 CONTINUE
   WRITE(6,2) J
   2 FORMAT(I8)
   STOP
   END

3. IDENTIFICATION DIVISION.
   PROGRAM-ID. 'HELLO'.
   ENVIRONMENT DIVISION.
   CONFIGURATION SECTION.
   SOURCE-COMPUTER. IBM-360.
   OBJECT-COMPUTER. IBM-360.
   SPECIAL-NAMES.
   CONSOLE IS CNSL.
   DATA DIVISION.
   WORKING-STORAGE SECTION.
   77 HELLO-CONST PIC X(12) VALUE 'HELLO,WORLD'.
   PROCEDURE DIVISION.
   000 DISPLAY.
   DISPLAY HELLO-CONST UPON CNSL.
   STOP RUN.

4. (defun fact (n)
   (do* ((i 1 (+ i 1)) (j 1 (* j i)))
        ((>= i n) j)))

5. □ ← */10

6. def factorial(n):
   result = 1

*Majority of the material has been gathered with help from the TAs of this course over the past few years.
for i in range(1, n+1):
    result *= i
return result

7. int fact (int n)
   {
       int i, j;
       j = 1;
       for (i=1; i<=n; i++)
           j *= i;
       return j;
   }

8. fact :: Int -> Int
    fact 1 = 1
    fact n = n * fact (n-1)

9. let rec fact n =
       if n==1
           then 1
           else n * fact (n-1);

10. fact(1, 1).
    fact(N, M) :- N > 1, fact (N-1, M1), M=M1*N.

11. counter=$1
    factorial=1
    while [ $counter -gt 0 ]
        do
            factorial=$(( $factorial * $counter ))
            counter=$(( $counter - 1 ))
        done
    echo $factorial

12. /factorial {
        dup 1 eq {}{
            dup 1 sub factorial mul
        } ifelse
    } def

13. function fac(n){
        return(n<2)?1:fac(n-1)*n;
    }

Exercise 2: Representation of numbers

1. How many values can a 1 bit integer take? What about 3 bits? What about n bits?
2. You’re building a fence 100 feet long, with posts every 10 feet. How many posts do you need?

Unsigned numbers

The sequence $\vec{a} \equiv a_{n-1} \cdots a_0$ of digits is interpreted as

$$[\vec{a}]_u \equiv \sum_{k=0}^{n-1} a_k 2^k$$
Two’s complement, AKA signed numbers

The sequence $\vec{a} \triangleq a_{n-1} \cdots a_0$ of digits is interpreted as

$$[\vec{a}]_{tc} \triangleq -a_{n-1}2^{n-1} + \sum_{k=0}^{n-2} a_k 2^k$$

3. What values can a natural number represented using $n$ bits take? What about a signed number?

4. Compute the following additions on 4 bit unsigned numbers:
   (a) $0010 + 0110$
   (b) $0101 + 1010$
   (c) $1011 + 1101$
   (d) $1010 + 0110$

One’s complement

The sequence $\vec{a} \triangleq a_{n-1} \cdots a_0$ of digits is interpreted as

$$[\vec{a}]_{tc} \triangleq \begin{cases} \sum_{k=0}^{n-2} a_k 2^k & \text{if } a_{n-1} = 0 \\ \sum_{k=0}^{n-2} (a_k - 1) 2^k & \text{otherwise} \end{cases}$$

5. How does one write $1$ using One’s complement? What about $-1$? How can you negate a number?

6. What is a huge drawback of this representation?

7. Using previous examples, build an algorithm to add two numbers in One’s complement. (Hint: the question is, how to handle the carry).

8. Why does your algorithm terminate?

9. What is printed by the Java program below?

```java
byte i = 101, j = 87, k = -101, l = -99;
byte m, n, o;
m = i+j; n = j+k; o = k+l;
System.out.println(m);
System.out.println(n);
System.out.println(o);
```

Exercise 3 : Representation of text

1. Decode the following ASCII string (written using hexadecimal codes)

```
64 6f 6e 27 74 20 70 61 6e 69 63
```

A few Unicode characters

<table>
<thead>
<tr>
<th>Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>U+000A</td>
<td>LINE FEED (LF)</td>
</tr>
<tr>
<td>U+0020</td>
<td>SPACE</td>
</tr>
<tr>
<td>U+0021</td>
<td>EXCLAMATION MARK</td>
</tr>
<tr>
<td>U+002C</td>
<td>COMMA</td>
</tr>
<tr>
<td>U+0030</td>
<td>DIGIT ZERO</td>
</tr>
<tr>
<td>U+0041</td>
<td>LATIN CAPITAL LETTER A</td>
</tr>
<tr>
<td>U+0061</td>
<td>LATIN SMALL LETTER A</td>
</tr>
</tbody>
</table>
2. What could be the shortcomings of UTF-32?

<table>
<thead>
<tr>
<th>UTF-8 encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>— U+0000 à U+007F : 0xxxxxxx</td>
</tr>
<tr>
<td>— U+0080 à U+07FF : 110xxxxx 10xxxxxx</td>
</tr>
<tr>
<td>— U+0800 à U+FFFF : 1110xxxx 10xxxxxx 10xxxxxx</td>
</tr>
<tr>
<td>— U-10000 à U-1FFFFF : 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx</td>
</tr>
</tbody>
</table>

3. Decode the following UTF-8 string

70 65 6E 20 70 69 6E 65 61 70 70 6C 65

4. Does UTF-8 have the same shortcomings as UTF-32? How and why?

5. When fetching the following webpage


my web browser displays:

Now remember that \((x_i)_{i \in I} \) converges to \(x\) if and only if every open subset \(U\) that contains \(x\) is such that \(x_i\) is eventually in \(U\). One obtains an equivalent definition by stating that every neighborhood \(A\) of \(x\) (i.e., in \(N\)) is such that \(x_i\) is eventually in \(A\). In other words, if and only if \(N\) is included in the convergence filter of the net.

However, the server projects.lsv.ens-cachan.fr sent to my browser the following (extract) of code:

Now remember that \(<em>x_i</em>\) \(i \in I, \sqsubseteq\) converges to \(x\) if and only if every open subset \(<em>U</em>\) that contains \(x\) is such that \(<em>x_i</em>\) is eventually in \(<em>U</em>\). One obtains an equivalent definition by stating that every neighborhood \(<em>A</em>\) of \(<em>x</em>\) (i.e., in \(<em>N</em>\)) is such that \(<em>x_i</em>\) is eventually in \(<em>A</em>\). In other words, if and only if \(<em>N</em>x\) is included in the convergence filter of the net.

How does it compare to Unicode?

6. Going back to the first example of HTML, the file started with:

```html
<?
$EXTRA_HEAD="antispam.html";
$ARG_BODY="onload="onLoad()"";
SETLANG("fr")
STYLEDPINFO();
HEAD("Conférences de rentrée 2015");
ADDTITLE("Conférences de rentrée 2015");
MKPAGEDPINFO();
?>
```

This is not HTML. What language is used? What does it compute?