As in the rest of the course, log denotes the binary logarithm function. We note H_2 the binary entropy function defined by $H_2(x) \coloneqq -x \log x - (1-x)\log(1-x)$. The two exercises are independent. The exam is marked out of a total of 100 points (50 points for each exercise).

A) Binary Asymmetric Channel (50 points)

Let X be a binary random variable and p_0, p_1 two real numbers in $[0;1]$. We consider the binary asymmetric channel $\mathrm{BAC}(p_0, p_1)$ that, given input X, defines a binary random variable output Y by
$\begin{cases} \Pr(Y = 1 \mid X = 0) = p_0; \\ \Pr(Y = 0 \mid X = 1) = p_1. \end{cases}$

1. (4 points) Draw the transition diagram for $\mathrm{BAC}(p_0, p_1)$.
2. (4 points) We let $\alpha = \Pr(X = 0)$. Express $H(Y)$ in terms of the binary entropy function H_2, α, p_0, and p_1.
3. (5 points) Compute $I(X;Y)$.
4. (5 points) What is the value of $I(X;Y)$ when $p_0 + p_1 = 1$? Explain the intuition behind this result.
5. (12 points) From now on, we assume $p_0 + p_1 < 1$. Let $z = 2^{\frac{H_2(p_0) - H_2(p_1)}{1 - p_0 - p_1}}$. Compute $\frac{\mathrm{d}I(X;Y)}{\mathrm{d}\alpha}$, the derivative of $I(X;Y)$ with respect to α, and express it in terms of z.
6. (12 points) Show that the capacity $C_{\mathrm{BAC}}(p_0, p_1)$ of the binary asymmetric channel is: $C_{\mathrm{BAC}}(p_0, p_1) = \log(1+z) - \log z + p_1 \log z - H_2(p_1)$. Verify that you obtain the capacity of the binary symmetric channel when $p_0 = p_1$.
7. (8 points) Alice is sending a secret key to Bob over a perfect, noiseless channel. Her message is thus a sequence of n i.i.d. uniformly distributed bits. Alice and Bob are not aware that Eve has access to secret information, through a binary asymmetric noisy channel $\mathrm{BAC}(p_0, p_1)$. We assume $p_0 < \frac{1}{2}$, $p_1 < \frac{1}{2}$. What proportion of the message between Alice and Bob can Eve expect to correctly guess?

B) TinyLang and Unexpectedness (50 points)

In this exercise, we will consider a minimal imperative programming language TINYLANG built using the 16 following symbols (keywords **`if`**, **`while`**, **`output`** are seen as one symbol each):

`0` `1` `v` `{` `}` `(` `)` `+` `*` `=` $\neq$ $\leftarrow$ **`if`** **`while`** **`output`** `;`

1. (4 points) Assuming a uniform prior on the symbols, propose an optimal uniquely decodable binary symbol code C for the symbols of TINYLANG.

   The programming language TINYLANG is built using the following constructs:

   - A test construct "**`if`** *`Expression`* `{` *`Instructions`* `}`" that executes a sequence of *`Instructions`* if the *`Expression`* evaluates to a non-zero value.
   - A loop construct "**`while`** *`Expression`* `{` *`Instructions`* `}`" that executes a sequence of *`Instructions`* as long as *`Expression`* evaluates to a non-zero value.
   - An *`Instruction`* is either a test construct, a loop construct, an assignment of the form "*`Variable`* $\leftarrow$ *`Expression`*", or an output instruction "**`output`** *`Expression`*". Within a sequence, *`Instructions`* are separated by a semicolon ("`;`") and are executed sequentially.

- An expression is a well-formed arithmetic expression on integers built using the "`+`" and "$*$" arithmetic operators, the "=" and "$\neq$" comparison operators, *`Variable`* names, *`Integer`* constants, and parentheses "(", ")". The "=" operator returns 1 if both operands are equal, 0 otherwise, and vice versa for "$\neq$".
- *`Integer`* constants are arbitrary nonnegative integers in binary, with no leading "`0`" except for the constant "`0`".
- *`Variable`* names are written as the letter "`v`" followed by an arbitrary nonnegative integer in binary, with no leading "`0`" except for the variable name "`v0`".

A program is a sequence of *Instructions*. The output of the program is the sequence of nonnegative integers produced by the **`output`** instructions, in the order they were produced. One assumes that all operations on integers are executed at an arbitrary precision, without any possible overflow.

As an illustration, here is a TINYLANG program $P_{12!}$ computing the factorial of twelve:

```
v0←1;v1←1;whilev0≠1101{v1←v1*v0;v0←v0+1;}outputv1
```

or, with spaces just for readability:

```
v0 ← 1;
v1 ← 1;
while v0≠1101 {
  v1 ← v1*v0;
  v0 ← v0+1;
}
output v1
```

2. (4 points) We extend C into a code on programs of TINYLANG by concatenating the codes of each symbol. What are the first 40 bits of $C(P_{12!})$?

3. (10 points) Let n be an arbitrary nonnegative integer n. Write a TINYLANG program $P_{1\ldots n}$ that generates the sequence $(1, 2, \ldots, n)$; this program should be as short as possible for large n. What is the size of $C(P_{1\ldots n})$ in bits in terms of n?

4. (8 points) For a sequence of positive integers $S = (s_1 \ldots s_n)$, we define the TINYLANG *Kolmogorov complexity of* S, $K_{\text{TINYLANG}}(S)$, as the minimum size of the encoding $C(P)$ of a TINYLANG program P with output the sequence S. Show that $K_{\text{TINYLANG}}(S) \leqslant 12n + 4 \log \prod_{i=1}^{n} s_i$.

5. (24 points) We consider a lottery where k numbers are drawn among n possible numbers $1 \ldots n$, the same number not being drawn twice (for example, disregarding the additional number, the Singaporean Toto lottery has $k = 6$ and $n = 49$). Order is irrelevant. This defines a random variable $X_{k,n}$ where each event is a draw (i.e., a set of k distinct numbers).
   a) (6 points) What is the entropy $H(X_{k,n})$ of $X_{k,n}$? Using Stirling's formula, $n! \sim \left(\frac{n}{\mathrm{e}}\right)^n \sqrt{2\pi n}$, show that $H(X_{k,n}) \sim -n \log \left(1 - \frac{k}{n}\right) + k \log \left(\frac{n}{k} - 1\right)$.
   b) (18 points) One defines an *unexpected* lottery draw as a sequence of k distinct integers among $1 \ldots n$ such that $K_{\text{TINYLANG}}(S) \ll H(X_{k,n})$ ("$\ll$" is read "significantly smaller than").
      i. (5 points) Give an example of unexpected sequence for some k, n.
      ii. (8 points) We now establish an upper bound on the probability for a sequence to be unexpected. Show that $\Pr\left(K_{\text{TINYLANG}}(S) \leqslant H(X_{k,n})/\gamma\right) \leqslant 2\binom{n}{k}^{-1+\frac{1}{\gamma}}$ for any constant $\gamma > 1$. What do you conclude?
      iii. (5 points) Discuss the intuitive interpretation of *unexpectedness*.

END OF PAPER