Q1. (6 marks)
a) Consider the sequence $(s_i)_{i \in N}$, where s_i is the number of bitstrings of length i that contain three consecutive 0s. Define this sequence recursively.
b) Consider the sequence $(t_i)_{i \in N}$, where t_i is the number of bitstrings of length i that contain the string 01. Define this sequence recursively.
c) Consider the sequence $(u_i)_{i \in Z^+}$, where u_i is the number of messages that can be transmitted in i microseconds using three different signals: one signal requires 1 microsecond for transmittal, the other two signals require 2 microseconds each for transmittal, and a signal in a message is followed immediately by the next signal. Define this sequence recursively.

The terms and elements of all the permutations and combinations considered here are integers.

Q1.

Using the library provided in Appendix, write in pseudocode the procedure
permutation nextPermutation2 (permutation p)

which returns the next n-permutation in lexicographic order after p, where n is the length of p (but p is not necessarily an n-permutation of 1..n). Return the empty string if p is the last n permutation. For example, return (−3,1,11,5) if p=(−3,1,5,11) and return () if p=(11,5,1,−3).

q4. Using the procedure above and the library provided in Appendix, write in pseudocode
void printPermutations (integer n)

which prints ALL the permutations of 1..n. The empty string must be printed first, then the 1-permutations, the 2-permutations, etc.

Appendix

void printCombination (combination c) Prints c.

integer cardinalCombination (combination c) Returns
the cardinal of c.

integer elementCombination (integer i, combination c)
Returns the i^{th} term of the increasing n-permutation of c, where n is the cardinal of c.

combination firstCombination (integer n) Returns
the combination 1..n.

combination nextCombination (integer n, combination c)
Assumes c is a combination of 1..n. Returns the next r-combination in lexicographic order after c, where r is the cardinal of c. Returns the empty set if c is the last r-combination of 1..n.

---

void printPermutation (permutation p) Prints
p.

integer lengthPermutation (permutation p) Returns
the length of p.

integer termPermutation (integer i, permutation p) Returns
the i^{th} term of p.

permutation firstPermutation (integer n) Returns
the permutation $(i)_{i \in 1..n}$.

permutation nextPermutation (permutation p)
Assumes p is an n-permutation of 1..n. Returns the next n-permutation in lexicographic order after p. Returns the empty string if p is the last n-permutation of 1..n.

permutation sortPermutation (permutation p)
Returns the permutation $(q_i)_{i \in 1..n}$ of 1..n, where n is the length of p, such that p_i is the q_i^{th} smaller term of p. For example, returns (3,2,4,1) if p=(5,−3, 99,−12).

permutation rearrangePermutation (permutation p, permutation q)
Assumes that q is an n-permutation of 1..n, where n is the length of p.
Returns the n-permutation whose i^{th} term is the q_i^{th} smaller term of p.

For example, returns (−3,5,−12,99) if p=(5,−3, 99,−12) and q=(2,3,1,4).

---

permutation combinationToPermutation (combination c)

Returns the increasing n-permutation of c, where n is the cardinal of c.