* 1. There is a set P = {'a', 'b', 'c', ..., 'z'} of small English letters. The task is to create table T with 8 columns. Each row of T will contain one 8-element subset of P and each cell in the row will contain exactly one letter. All 8-element subsets of P will be listed in T and no subset will be listed twice. Find the size of the table and determine if it is possible for your personal computer to fill it completely in less than 1 second.

* 2. Write a pseudocode of a function which will print out all unempty subsets of the set $\{0, 1, 2, ..., n-1\}$.

* 3. Consider permutations of the set M = $\{1, 2, 3, ..., n\}$, $n > 4$. A permutation p of M is said to be *cheerful* if the following holds: $p(3) \in \{3, n\}$; $p(n) \in \{3, n\}$; $p(1) = 1$; $p(2) = 2$; $p(i) \in \{4, ..., n-1\}$ for $i = 3, 4, ..., n-1$.
Find the number of all *cheerful* permutations of M.

* 4. Suppose that every element of Gray code G^n (i.e. n-touple of 0's and 1's) is stored in a character array of length n. Write a pseudocode of a function which prints out the complete Gray code G^n.

* 5. Set M contains 98 elements. Each permutation of M is ranked by an unique integer in the range from 0 to $98!-1$. The program represents each permutation by its rank. We know that in any moment the program will store at most 100 permutations of M and therefore it will allocate static memory for just 100 ranks of those permutations. What is the minimum number of bits needed to store any 100 of ranks?

6. A sequence P = (000, 001, 011, 110, 111, 101, 100) represents a Gray Code G^3. Two finite sequences are said to be equivalent if:
1. A reversed is equal to B, or
2. Left or right rotation of A by any number of positions is equal to B, or
3. There exists sequence C equivalent to A nad B.

Find an 8-element sequence Q which is a Gray code and which is not equivalent to P. Here we define Gray code to be any a binary system where each two neighbour codes differ in only one bit.

7. Consider all k-element subsets of set M = $\{1, 2, 3, ..., n\}$, $1 \le k \le n$. There exists an algorithm which transforms a list of slements of one subset into a list of lements of the next subset in the lexicographical ordering of all k-element subsets of M. Your task is to design a reverse algorithm, i.e. an algorithm which which transforms a list of slements of one subset into a list of lements of the previous subset in the lexicographical ordering of all k-element subsets of M. Will the asymptotic complexity of both transformations be the same?

8. Consider permutations of set M = $\{1, 2, 3, ..., n\}$. We define a cycle of length k in permutation p to be a set A = $\{a_1, a_2, ..., a_k\} \subseteq$ M, for which holds:
$1 \le a_1 < a_2 < ... < a_k \le n$; $p(a_j) = a_{j+1}$ for $1 \le j < k$; $p(a_k) = a_1$.
Determine the number of such permutations of M which contain exactly two cycles and moreover the length of one cycle is 4 and the lenght of the other cycle is $n-4$.

9. Rank of a permutation π of set N = $\{0, 1, 2, ..., n-1\}$ is the index of π in the list of all permutations of N. The list is sorted in increasing lexicographical order and is indexed from 0. Write a pseudocode of a function which will print out the permutation which has rank $n!/2$ and will do it in time proportional to n. Suppose $n \ge 2$.

10 . Consider all permutations of set M = $\{1, 2, 3, ..., n\}$, $1 \le k \le n$. There exists an algorithm which transforms one permutation into the permutation which is next the lexicographical ordering of all permutations of M. Your task is to design a reverse algorithm, i.e. an algorithm which transforms one permutation into the permutation which is previous the lexicographical ordering of all permutations of M. Will the asymptotic complexity of both transformations be the same?

11. Permutation p of set M is called derangements, if for each $k \in$ M holds
$1 \le k \le n \Rightarrow p(k) \ne k$. Write down all derangements of the set $\{1, 2, 3, 4\}$.
Find 1000000-th element in the lexicographical ordering of derangements of set $\{1, 2, 3, ..., 20\}$.