```
> with(combinat);
[Chi, bell, binomial, cartprod, character, choose, composition, conjpart, decodepart,        (1)
    encodepart, eulerian1, eulerian2, fibonacci, firstcomb, firstpart, firstperm, graycode,
    inttovec, lastcomb, lastpart, lastperm, multinomial, nextcomb, nextpart, nextperm,
    numbcomb, numbcomp, numbpart, numbperm, partition, permute, powerset, prevcomb,
    prevpart, prevperm, randcomb, randpart, randperm, rankcomb, rankperm, setpartition,
    stirling1, stirling2, subsets, unrankcomb, unrankperm, vectoint]
> ?permute;
> permute(4);
[[1, 2, 3, 4], [1, 2, 4, 3], [1, 3, 2, 4], [1, 3, 4, 2], [1, 4, 2, 3], [1, 4, 3, 2], [2, 1, 3, 4], [2, 1, 4,        (2)
    3], [2, 3, 1, 4], [2, 3, 4, 1], [2, 4, 1, 3], [2, 4, 3, 1], [3, 1, 2, 4], [3, 1, 4, 2], [3, 2, 1, 4],
    [3, 2, 4, 1], [3, 4, 1, 2], [3, 4, 2, 1], [4, 1, 2, 3], [4, 1, 3, 2], [4, 2, 1, 3], [4, 2, 3, 1], [4, 3,
    1, 2], [4, 3, 2, 1]]
> numbperm(4, 4);
                                        24                                        (3)
> permute(5, 2);
[[1, 2], [1, 3], [1, 4], [1, 5], [2, 1], [2, 3], [2, 4], [2, 5], [3, 1], [3, 2], [3, 4], [3, 5], [4, 1],        (4)
    [4, 2], [4, 3], [4, 5], [5, 1], [5, 2], [5, 3], [5, 4]]
> numbperm(5, 2);
                                        20                                        (5)
> permute([a, a, c, d], 3);
[[a, a, c], [a, a, d], [a, c, a], [a, c, d], [a, d, a], [a, d, c], [c, a, a], [c, a, d], [c, d, a], [d, a,        (6)
    a], [d, a, c], [d, c, a]]
> numbperm([a, a, c, d], 3);
                                        12                                        (7)
> permute([a, b, f]);
              [[a, b, f], [a, f, b], [b, a, f], [b, f, a], [f, a, b], [f, b, a]]              (8)
> permute([a, b, c, d], 2);
   [[a, b], [a, c], [a, d], [b, a], [b, c], [b, d], [c, a], [c, b], [c, d], [d, a], [d, b], [d, c]]   (9)
>
> choose([a, b, f]);
                  [[ ], [a], [b], [a, b], [f], [a, f], [b, f], [a, b, f]]                  (10)
> choose([a, b, c, d], 2);
                    [[a, b], [a, c], [a, d], [b, c], [b, d], [c, d]]                    (11)
> nops(choose([a, b, c, d], 2));
                                        6                                        (12)
> orderedLists := proc(S, k)
  local i, L : L := NULL :
  for i from 1 to nops(S) do
  L := L, seq(S[i], j = 1 ..k)
   od: RETURN(permute([L], k)) end;
orderedLists := proc(S, k)                                                       (13)
    local i, L;
    L := NULL;
    for i to nops(S) do L := L, seq(S[i], j = 1 ..k) end do;
    RETURN(combinat:-permute([L], k))
end proc
> orderedLists([a, b, c, d], 2);
[[a, a], [a, b], [a, c], [a, d], [b, a], [b, b], [b, c], [b, d], [c, a], [c, b], [c, c], [c, d], [d, a],     (14)
    [d, b], [d, c], [d, d]]
> orderedLists([a, b, c, d, e, f, g, h, i, j], 2);
[[a, a], [a, b], [a, c], [a, d], [a, e], [a, f], [a, g], [a, h], [a, i], [a, j], [b, a], [b, b], [b, c],     (15)
    [b, d], [b, e], [b, f], [b, g], [b, h], [b, i], [b, j], [c, a], [c, b], [c, c], [c, d], [c, e], [c, f],
    [c, g], [c, h], [c, i], [c, j], [d, a], [d, b], [d, c], [d, d], [d, e], [d, f], [d, g], [d, h], [d, i],
    [d, j], [e, a], [e, b], [e, c], [e, d], [e, e], [e, f], [e, g], [e, h], [e, i], [e, j], [f, a], [f, b],
    [f, c], [f, d], [f, e], [f, f], [f, g], [f, h], [f, i], [f, j], [g, a], [g, b], [g, c], [g, d], [g, e],
    [g, f], [g, g], [g, h], [g, i], [g, j], [h, a], [h, b], [h, c], [h, d], [h, e], [h, f], [h, g], [h, h],
    [h, i], [h, j], [i, a], [i, b], [i, c], [i, d], [i, e], [i, f], [i, g], [i, h], [i, i], [i, j], [j, a], [j, b],
    [j, c], [j, d], [j, e], [j, f], [j, g], [j, h], [j, i], [j, j]]
> permute(3, 2);
                    [[1, 2], [1, 3], [2, 1], [2, 3], [3, 1], [3, 2]]                    (16)
> nops(permute(3, 0));
                                        1                                           (17)
> nops(permute(3, 1));
                                        3                                           (18)
> nops(permute(3, 2));
                                        6                                           (19)
> nops(permute(3, 3));
                                        6                                           (20)
> nops(choose(3, 0));
                                        1                                           (21)
> nops(choose(3, 1));
                                        3                                           (22)
> nops(choose(3, 2));
                                        3                                           (23)
> nops(choose(3, 3));
                                        1                                           (24)
> permute([H, H, H, H, T, T], 5);
[[H, H, H, H, T], [H, H, H, T, H], [H, H, H, T, T], [H, H, T, H, H], [H, H, T, H, T], [H, H, T, T,     (25)
    H], [H, T, H, H, H], [H, T, H, H, T], [H, T, H, T, H], [H, T, T, H, H], [T, H, H, H, H], [T, H,
    H, H, T], [T, H, H, T, H], [T, H, T, H, H], [T, T, H, H, H]]
> nops(permute(4, 0));
                                        1                                           (26)
> nops(permute(4, 1));
                                        4                                           (27)
> nops(permute(4, 2));
                                        12                                          (28)
```

```
> nops(permute(4, 3));
                              24                              (29)
> nops(permute(4, 4));
                              24                              (30)
> nops(choose(4, 0));
                               1                              (31)
> nops(choose(4, 1));
                               4                              (32)
> nops(choose(4, 2));
                               6                              (33)
> nops(choose(4, 3));
                               4                              (34)
> nops(choose(4, 4));
                               1                              (35)
> for i from 0 to 3 do
   S := choose(3, i);
    print(S);
   od
                          S := [[ ]]
                            [[ ]]
                    S := [[1], [2], [3]]
                      [[1], [2], [3]]
               S := [[1, 2], [1, 3], [2, 3]]
                 [[1, 2], [1, 3], [2, 3]]
                      S := [[1, 2, 3]]
                         [[1, 2, 3]]                          (36)
                              10                              (37)
> for i from 0 to 3 do
   S := numbperm(3, i);
    print(S);
   od:


                               1
                               3
                               6
                               6                              (38)
> 
> 
```