```
/* This is a collection of functions that deal with (abstract
simplicial) complexes.  ASCs are given as sets of sets - the complete
ASC is the set of all nonempty subsets of the sets.  */

//Vertex set
V:=func<K | &join({s : s in K})>;

//Dimension
dim:=func<K| Max({#s : s in K})-1>;

//Euler characteristic of complex K
Euler:=function(K);
vertexset:=V(K);
d:=Max({ #s : s in K})-1;
return
&+[(-1)^k*#{s : s in Subsets(vertexset,k+1) |
exists{t : t in K | s subset t}} : k in [0..d]];
end function;

//Join of two complexes
JOIN:=function(K1,K2);
m1:=Max(&join(K1));
m2:=Min(&join(K2));
K2:={{m1+1+j-m2 : j in s} : s in K2};
return
{ s1 join s2 : s1 in K1, s2 in K2};
end function;

//Suspension of complex K is S^0*K
SUSP:=func<K|JOIN({{0},{1}},K)>;

//The n-sphere
S:=func<n|Subsets({IntegerRing() | 1..n+2},n+1)>;

//n-fold suspension
SUSP:=func<K,n|JOIN(S(n-1),K)>;

/*
Surfaces
*/

S2:=Subsets({IntegerRing() | 1..4},3);

P2:={
{1,2,4}, {1,3,4}, {3,4,6}, {2,3,6}, {1,2,6}, {1,5,6}, {1,3,5}, {2,3,5},
{2,4,5}, {4,5,6}};
// http://infoshako.sk.tsukuba.ac.jp/~HACHI/math/library/projective_eng.html

T2:={
{1,2,6}, {2,3,7}, {1,3,4},
{1,4,6}, {2,6,7}, {3,4,7},
{4,6,8}, {6,7,9}, {4,5,7},
{4,5,8}, {6,8,9}, {5,7,9},
{2,5,8}, {3,8,9}, {1,5,9},
{1,2,5}, {2,3,8}, {1,3,9}};
```

```
//Connected sum of two surfaces
CSUM:=function(K1,K2);
m1:=Max(&join(K1));
m2:=Min(&join(K2));
K2:={{m1+1+j-m2 : j in s} : s in K2};
s1:=Rep(K1);
s2:=Rep(K2);
x:=SetToSequence(s1);
y:=SetToSequence(s2);
return
(K1 diff {s1}) join (K2 diff {s2})
join { {y[1],x[2],x[3]}, {x[1],y[2],x[3]}, {x[1],x[2],y[3]} }
join { {x[1],y[2],y[3]}, {y[1],x[2],y[3]}, {y[1],y[2],x[3]} };
end function;


//Orientable and nonorientable surfaces
M:=
func<g| [k eq 1 select T2 else CSUM(Self(k-1),T2) : k in [1..g]][g]>;
N:=
func<h| [k eq 1 select P2 else CSUM(Self(k-1),P2) : k in [1..h]][h]>;


/*
Homology
*/

Z:=IntegerRing();

//Put the elements of sequence S of integers in increasing order
order:=func<S|
[k eq 1 select Min(S) else Min(S diff {Self(i) : i in [1..k-1]}) : k
in [1..#S]]>;

//Compute the rth homology group of K with coefficients in R
H:=function(K,r,R);
d:=dim(K);
assert r in [0..d];
sk:=[[order(s) : s in Setseq(&join{ Subsets(t,k+1) : t in K })] :
k in [0..d]]; //sk[n] is the set of n+1-simplices
D:=func<n|
Matrix(R,#sk[n+1],#sk[n],
[<i,Index(sk[n],Remove(sk[n+1][i],j)), (-1)^j> :
j in [1..n+1], i in [1..#sk[n+1]]])>;  //the differential in the chain complex
return
case<r |
d : quo<Kernel(D(d))|[]>,
0 : quo<RSpace(R,#sk[1])|RowSequence(D(1))>,
default : quo<Kernel(D(r))|RowSequence(D(r+1)) > >;
end function;


 /*
//Examples
```

```
[Euler(SUSP(K),1) eq 2-Euler(K) : K in {S2,P2,T2}];
[Euler(JOIN(K1,K2)) eq Euler(K1)+Euler(K2)-Euler(K1)*Euler(K2) : K1,K2
in {S2,P2}];

[Euler(N(h)) eq 2-h : h in [1..5]];
[Euler(M(g)) eq 2-2*g : g in [1..5]];

[Euler(CSUM(K1,K2)) eq Euler(K1)+Euler(K2)-2 : K1,K2 in {S2,P2,N(2),N(3)}];

[Euler(CSUM(M(g),M(h))) eq Euler(M(g+h)) : h,g in [1..2]];
[Euler(CSUM(M(g),N(h))) eq Euler(N(2*g+h)) : h,g in [1..3]];
[Euler(CSUM(N(g),N(h))) eq Euler(N(g+h)) : h,g in [1..3]];

H(M(3),2,GF(2));
H(N(4),1,Z);
[Dimension(H(M(2),i,GF(2))) : i in [0..2]];
[Dimension(H(SUSP(M(2),1),i,GF(2))) : i in [0..3]];
[Dimension(H(SUSP(M(2),2),i,GF(2))) :  i in [0..4]];
[Dimension(H(JOIN(M(2),N(3)),i,GF(2))) : i in [0..5]];

H(JOIN(M(2),N(3)),4,Z);
H(JOIN(M(2),N(3)),4,GF(2));
*/

/* JUNK


D:=function(n,sk);
return
Matrix(Z,#sk[n+1],#sk[n],
&cat[
[exists(k){k : k in [1..n+1] | j eq Index(sk[n],Remove(sk[n+1][i],k))} select
(-1)^k else 0 : j in [1..#sk[n]]] : i in [1..#sk[n+1]]]);
end function;


//SK[n] is the indexed set of  n-simplices in K
SK:=func<K|
[[order(s) : s in Setseq(&join{ Subsets(t,k+1) : t in K })] :
k in [0..dim(K)]]>;

//matrix for differential n -> n-1
D:=func<n,sk|
Matrix(Z,#sk[n+1],#sk[n],
&cat[[<i,Index(sk[n],Remove(sk[n+1][i],j)), (-1)^j> : j in [1..n+1]] :
i in [1..#sk[n+1]]])>;


//Compute the homology of complex K in degree r
homology:=function(K,r);
sk:=SK(K);
d:=dim(K);
return
case<r |
d : quo<Kernel(D(d,sk))|Image(Matrix(Z,1,#(sk[d+1]),[]))>,
```

```
0 : quo<Kernel(Matrix(Z,#(sk[1]),1,[]))|Image(D(1,sk))>,
default : quo<Kernel(D(r,sk))|Image(D(r+1,sk))> >;
end function;



mfld(CSUM(M(2),N(3)));
mfld(JOIN(M(2),N(3)));
mfld(SUSP(M(6)));
mfld(JOIN(SUSP(M(6)),SUSP(N(2))));


//Is the ASC a manifold (each (d-1) dim simplex lies in exactly two
//d-dimensional simplices
mfld:=function(K);
d:=dim(K);
top:=&join({ Subsets(t,d+1) : t in K});
hyp:=&join({ Subsets(t,d) : t in K});
return
forall{s : s in hyp | #{t : t in top | s subset t} eq 2};
end function;


*/
```