```
Define Mult(I)
  If Type(I) = IDEAL Then Mult := Multiplicity(CurrentRing()/I)
    Else Mult := 0;
  End;
  Return(Mult);
End;

DegWithQ:= [Mult(K) | K In WithQ(APO)]; DegWithQ;
[1, 48, 1, 1, 8, 8, 8]
-------------------------------
DegWithoutQ:= [Mult(K) | K In WithoutQ]; DegWithoutQ;
[0, 44, 0, 0, 7, 7, 7]
-------------------------------
Degrees:= DegWithQ - DegWithoutQ; Degrees;
[1, 4, 1, 1, 1, 1, 1]
-------------------------------
ArithDeg:= Sum(Degrees); ArithDeg;
10
-------------------------------
```

The contribution of a single irreducible component to the arithmetic degree is only well-defined after choosing an ordering of the irreducible components. However, the total contribution from a given *associated* prime is well defined.

B.2 Solutions for Day 2

Solution B.2.1 *By (Bahman Engheta, Leah Gold*, Ed Mosteig), (Kimberly Presser*), (Gerhard Quarg*), and (Carolyn Yackel*)*

The following CoCoA code uses the function `FVector`, defined as in Solution B.1.3. We assume this function has already been entered into CoCoA.

```
Use S ::= Q[x[1..4]y[1..4]z[1..4]];
-- the polarization appears next:
IDelta := Ideal(x[1]x[2]x[3]x[4],y[1]y[2]y[3]y[4],z[1]z[2]z[3]z[4],
x[1]x[2]x[3]y[1]y[2]z[1],x[1]y[1]y[2]y[3]z[1]z[2],
x[1]x[2]y[1]z[1]z[2]z[3]);
FVector(IDelta);
[1, 12, 66, 220, 492, 768, 837, 609, 264, 51]
-------------------------------
```

Thus, Δ has 12 vertices, 66 edges, etc.

Gerhard Quarg found another way to calculate the f-vector. First calculate the Hilbert series.

```
Poincare(S/IDelta);
(1 + 3x[1] + 6x[1]^2 + 10x[1]^3 + 12x[1]^4 + 12x[1]^5 + 7x[1]^6) /
(1-x[1])^9
-------------------------------
```

Now use the formula

$$H(t) = \frac{1}{(1-t)^{12}} \sum_{i=0}^{12} f_{i-1}\, t^i (1-t)^{12-i}.$$

Multiply by $(1-t)^{12}$ to get the numerator we need and compute the f vector using the CoCoA function `GenRepr`. The indeterminate `x[1]` plays the role of t.

```
F := (1-x[1])^12(1 + 3x[1] + 6x[1]^2 + 10x[1]^3 + 12x[1]^4 +
12x[1]^5 + 7x[1]^6) / (1-x[1])^9;
GenRepr(F,Ideal([x[1]^I (1-x[1])^(12-I) | I In 0..12]));
[1, 12, 66, 220, 492, 768, 837, 609, 264, 51, 0, 0, 0]
-------------------------------
```

Solution B.2.2 *By Kimberly Presser*

```
Define Generic(I)
Help 'The CurrentRing() should be Q[x[1..N]] where N is the number of
      indeterminates.';
J:=I;
N:=NumIndets();
   For K:=1 To N Do
     F:=DensePoly(1);
     L:=Randomized(F);
     J:=Subst(J,x[K],L);
   End;
   Gin:=LT(J);
   Return Gin;
End;

Use R::=Q[x[1..3]],Lex;
M:=Ideal(x[1]^4,x[2]^4,x[3]^4,x[1]^3x[2]^2x[3],x[1]x[2]^3x[3]^2,
         x[1]^2x[2]x[3]^3);
G:=Generic(M);
G;
Ideal(x[1]^4, x[1]^3x[2], x[1]^3x[3], x[1]^2x[2]^3, x[1]^2x[2]^2x[3],
x[1]^2x[2]x[3]^2, x[1]^2x[3]^4, x[1]x[2]^5, x[1]x[2]^4x[3],
x[1]x[2]^3x[3]^2, x[1]x[2]^2x[3]^3, x[1]x[2]x[3]^4, x[1]x[3]^5,
x[2]^7, x[2]^6x[3], x[2]^5x[3]^2, x[2]^4x[3]^3, x[2]^3x[3]^4,
x[2]^2x[3]^5, x[2]x[3]^6, x[3]^7)
```

```
-------------------------------
Res(R/G);
0 --> R(-6)(+)R^2(-7)(+)R^6(-8)(+)R^7(-9) -->
R^3(-5)(+)R^5(-6)(+)R^13(-7)(+)R^15(-8) -->
R^3(-4)(+)R^3(-5)(+)R^7(-6)(+)R^8(-7) --> R
-------------------------------
```

Solution B.2.3

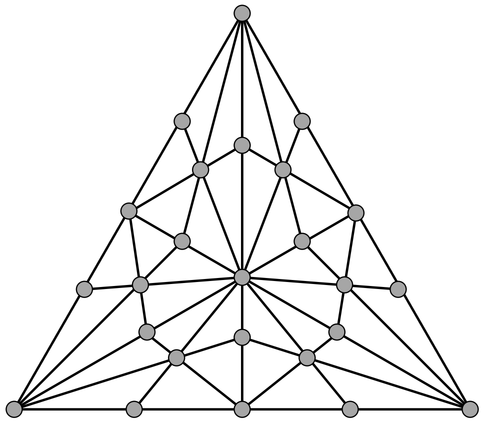
Clearly we will not be printing the solution to this one. However, the reader may find the following command useful.

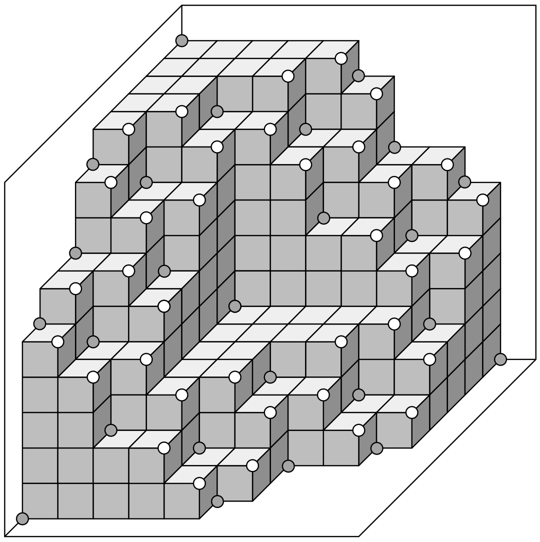
```
RandMon(N,Exp) := [LogToTerm([Rand(0,Exp) | L In 1..NumIndets()])
                   | M In 1..N];
```

It generates `N` random monomials with exponents between `0` and `Exp`.

Solution B.2.4 *By Burckhard Zimmerman*

The second barycentric subdivision of a triangle is at left below, and resolves the generic ideal with the staircase at right. Recall that the tree ideal (Section 5.3) has cellular minimal resolution supported on the *first* barycentric subdivision of a simplex. The above staircase is constructed roughly by arranging a bunch of tree ideals around a central inner corner. This arrangement is easiest to see by looking at the six hexagonal groups of white dots = irreducible components. [The first author of these Lectures was quite surprised when Mr. Zimmerman found precisely the same solution to this problem as he did. Most people think that the staircase is aesthetically more pleasing upside-down, representing the cogeneric Alexander dual.]





Solution B.2.5

CoCoA's standard use of `Poincare` is to give the $\mathbb{Z}$-graded Hilbert-Poincare series of a quotient S/M in reduced terms, so that the denominator is $(1-\text{variable})^d$, where $d =$

$\dim(S/M)$. If we could somehow convince CoCoA to give us the multigraded Hilbert series, the Scarf complex would consist of the monomials in the numerator. It turns out that CoCoA does have a feature for using `Poincare` when the variables have linearly independent *weights*, and this allows CoCoA to use finer gradings. Unfortunately, the first row of the `WeightsMatrix` must consist of positive integers, so we are not allowed to assign the weight $\mathbf{e}_i \in \mathbb{Z}^n$ to the i^{th} variable. As a meager substitute, we add a new homogenizing variable at the top of the list, and proceed to forget about it later. In our case, we need 4 variables a, b, c, and d, so we use t as the "zeroth" variable:

```
Use S ::= Q[t,a,b,c,d], Weights(Mat([[1,1,1,1,1],
                                     [0,1,0,0,0],
                                     [0,0,1,0,0],
                                     [0,0,0,1,0],
                                     [0,0,0,0,1]]));
M:=Ideal(a^5,b^5,c^5,d^5,ab^2c^3d^4,a^2b^3c^4d,a^3b^4cd^2,a^4bc^2d^3);
P:=Poincare(S/M);
```

It looks like `P` is a rational function, but really it's a list of coefficients and exponent vectors, as the command

```
Describe(P);
```

demonstrates. The elements of Δ_M are uniquely determined by their vector labels, which are obtained from the exponent vectors in the numerator of the Hilbert series by leaving off the exponent of `t`:

```
N := [Tail(X[2]) | X In P[1]];
N;
[[5, 5, 5, 1], [5, 5, 4, 2], [5, 4, 4, 3], [5, 3, 5, 3], [4, 4, 4, 4],
 [3, 5, 4, 4], [4, 3, 5, 4], [2, 5, 5, 4], [5, 5, 1, 5], [5, 4, 2, 5],
 [4, 4, 3, 5], [3, 5, 3, 5], [5, 1, 5, 5], [4, 2, 5, 5], [1, 5, 5, 5],
 [5, 5, 5, 0], [5, 5, 4, 1], [5, 4, 4, 2], [5, 3, 4, 3], [4, 4, 4, 3],
 [4, 3, 5, 3], [4, 4, 3, 4], [3, 5, 3, 4], [4, 3, 4, 4], [3, 4, 4, 4],
 [2, 5, 4, 4], [4, 2, 5, 4], [1, 5, 5, 4], [5, 5, 0, 5], [5, 4, 1, 5],
 [4, 4, 2, 5], [3, 4, 3, 5], [5, 0, 5, 5], [4, 1, 5, 5], [0, 5, 5, 5],
 [5, 3, 5, 1], [3, 5, 4, 2], [5, 4, 2, 3], [4, 3, 4, 3], [5, 1, 5, 3],
 [3, 4, 3, 4], [2, 3, 5, 4], [3, 5, 1, 5], [4, 2, 3, 5], [1, 5, 3, 5],
 [5, 3, 4, 1], [2, 5, 5, 1], [5, 5, 1, 2], [3, 4, 4, 2], [4, 4, 2, 3],
 [4, 1, 5, 3], [4, 2, 3, 4], [1, 5, 3, 4], [2, 3, 4, 4], [3, 4, 1, 5],
 [5, 1, 2, 5], [1, 2, 5, 5], [2, 5, 4, 1], [5, 4, 1, 2], [1, 2, 5, 4],
 [4, 1, 2, 5], [2, 3, 5, 1], [3, 5, 1, 2], [5, 1, 2, 3], [1, 2, 3, 5],
 [5, 5, 0, 0], [5, 0, 5, 0], [0, 5, 5, 0], [2, 3, 4, 1], [3, 4, 1, 2],
 [4, 1, 2, 3], [1, 2, 3, 4], [5, 0, 0, 5], [0, 5, 0, 5], [0, 0, 5, 5],
 [5, 0, 0, 0], [0, 5, 0, 0], [0, 0, 5, 0], [0, 0, 0, 5], [0, 0, 0, 0]]
```

If you want to know how the faces of Δ_M are described as least common multiples of minimal generators, just determine, for each element `N[I]` in the list `N`, which generators of `M` divide the monomial $(a,b,c,d)^{\texttt{N[I]}}$:

```
G := Gens(M);
D := [[J In 1..Len(G) | Mod(LogToTerm(Concat([0],N[I])),G[J]) = 0]
       | I In 1..Len(N)];
D;
[[1, 2, 3, 6], [1, 2, 6, 7], [1, 6, 7, 8], [1, 3, 6, 8], [5, 6, 7, 8],
 [2, 5, 6, 7], [3, 5, 6, 8], [2, 3, 5, 6], [1, 2, 4, 7], [1, 4, 7, 8],
 [4, 5, 7, 8], [2, 4, 5, 7], [1, 3, 4, 8], [3, 4, 5, 8], [2, 3, 4, 5],
 [1, 2, 3], [1, 2, 6], [1, 6, 7], [1, 6, 8], [6, 7, 8], [3, 6, 8],
 [5, 7, 8], [2, 5, 7], [5, 6, 8], [5, 6, 7], [2, 5, 6], [3, 5, 8],
 [2, 3, 5], [1, 2, 4], [1, 4, 7], [4, 7, 8], [4, 5, 7], [1, 3, 4],
 [3, 4, 8], [2, 3, 4], [1, 3, 6], [2, 6, 7], [1, 7, 8], [1, 3, 8],
 [3, 5, 6], [2, 4, 7], [4, 5, 8], [2, 4, 5], [2, 3, 6], [1, 2, 7],
 [1, 4, 8], [3, 4, 5],   [6, 8], [5, 7], [1, 6], [6, 7], [7, 8],
 [3, 8], [5, 8], [2, 5], [5, 6], [4, 7], [2, 6], [1, 7], [3, 5],
 [4, 8], [3, 6], [2, 7], [1, 8], [4, 5], [1, 2], [1, 3], [2, 3],
 [1, 4], [2, 4], [3, 4], [6], [7], [8], [5], [1], [2], [3], [4], [ ]]
```

Don't forget about the zeroth variable `t`, which is the reason for the `Concat`! This Scarf complex is the Schlegel diagram for the 4-dimensional cross-polytope (higher-dimensional analog of the octahedron), which can be expressed abstractly as the convex hull of the points $\pm\mathbf{e}_i$ for $i \in \{1, 2, 3, 4\}$.

Solution B.2.6 *By Carolyn Yackel*

All we need is the exponent vectors on the facets of the Scarf complex from the previous exercise. Continuing the CoCoA session from there, we want the elements of `N` whose corresponding element of `D` has length 4:

```
Irr := [N[I] | I In 1..Len(N) And Len(D[I]) = 4];
Irr;
[[5, 5, 5, 1], [5, 5, 4, 2], [5, 4, 4, 3], [5, 3, 5, 3], [4, 4, 4, 4],
 [3, 5, 4, 4], [4, 3, 5, 4], [2, 5, 5, 4], [5, 5, 1, 5], [5, 4, 2, 5],
 [4, 4, 3, 5], [3, 5, 3, 5], [5, 1, 5, 5], [4, 2, 5, 5], [1, 5, 5, 5]]
```

These $15 = 16 - 1 = 2^4 - 1$ facets of the cross-polytope correspond to all but one of the vertices of the hypercube, which is polar to the cross-polytope.

Solution B.2.7 *By Carolyn Yackel*

We may assume the ideal is generic, since deformation only makes the Betti numbers go up, and the number of irreducible components of an artinian ideal is the last nonzero Betti number. An upper bound for the number of irreducible components of a monomial ideal in n variables with r generators is given by one less than the number of facets of the cyclic n-polytope with r vertices. By a theorem of G. Agnarsson [1], this upper bound is attained if $n \leq 3$ or $n = 4$ and $r \leq 12$. Thus the upper bound 34 for $n = 4$ and $r = 10$ is attained by some monomial ideal. For example, leave out the last two generators of the ideal in Example 4.15.

Solution B.2.8

This is best done by the reader, following the model of Example 4.12.

B.3 Solutions for Day 3

Solution B.3.1 *By (Abdul Jarrah*), (Bahman Engheta, Leah Gold*, and Ed Mosteig)*

Given a monomial ideal I, let $\mathbf{a}_I$ be the exponent on the least common multiple of the minimal generators of I, so $I^\vee = I^{[\mathbf{a}_I]}$. It follows directly from the definition of Alexander dual ideal that $\mathbf{a}_{I^\vee} \preceq \mathbf{a}_I$. The definition also implies that if $\mathbf{a} \neq \mathbf{a}'$ are both $\succeq \mathbf{a}_I$, then $I^{[\mathbf{a}]} \neq I^{[\mathbf{a}']}$, because their corresponding generators are different. Therefore, since $(I^{[\mathbf{a}_I]})^{[\mathbf{a}_I]} = I$, it follows that $(I^\vee)^\vee = I$ if and only if $\mathbf{a}_I = \mathbf{a}_{I^\vee}$.

Claim B.1 $\mathbf{a}_I = \mathbf{a}_{I^\vee}$ *if and only if for each* $i \in \operatorname{supp}(\mathbf{a}_I)$ *there is a minimal generator* $\mathbf{x}^\mathbf{b}$ *of* I *with* $b_i = 1$.

Proof: $\Longrightarrow$: Let a_i and $b_i^\vee$ be the i^{th} coordinates of $\mathbf{a}_I$ and $\mathbf{a}_I \smallsetminus \mathbf{b}$. If there is some $\mathbf{x}^\mathbf{b}$ with $b_i = 1$, then $b_i^\vee = a_i$ by definition, so there is an irreducible component of $I^\vee$ with $x_i^{a_i}$ as one of its minimal generators. It follows from the algorithm in Section 0.6 for computing irreducible decompositions that a_i is the exponent on x_i in some minimal generator of $I^\vee$.

$\Longleftarrow$: If $x_i^{a_i}$ is the power of x_i in some minimal generator of $I^\vee$, then x_i is a minimal generator of some irreducible component of I, because $(I^\vee)^{[\mathbf{a}_I]}$ always equals I (Corollary 6.15). As before, this implies that x_i appears with exponent 1 in some minimal generator of I. □

Solution B.3.2

This follows from the definitions, using the following two observations.

1. $\operatorname{supp}(\mathbf{a} \smallsetminus \mathbf{b}) = \mathbf{1} \smallsetminus \operatorname{supp}(\mathbf{b}) = \operatorname{supp}(\mathbf{b})$.
2. $\sqrt{\bigcap \mathfrak{m}^\mathbf{b}} = \bigcap \sqrt{\mathfrak{m}^\mathbf{b}} = \bigcap \mathfrak{m}^{\operatorname{supp}(\mathbf{b})}$.

Solution B.3.3 *By Robert Forkel*

The function `AlexDual2` below is an implementation of the one-step algorithm [23, Theorem 2.1]. It tends to work faster in computer algebra systems (such as `Macaulay 2` [18]) designed for calculating kernels of matrices. The remaining two functions `AlexDual3` and `AlexDual4` are straightforward, and work faster in programs (like CoCoA) which deal well with monomials. Two examples and speed tests are given at the end.

```
Define AlexDual2(...)
  I:=Minimalized(ARGV[1]);
  If Len(ARGV)=2 Then A:=ARGV[2]
  Else A:=Log(LCM(Gens(I)));
  End;
```