- Polynomial SingleMult(poly, coef, expon) ::= return the polynomial poly.coef. x^{expon}

- Polynomial Add(poly1, poly2) ::= return the polynomial poly1+poly2
- Polynomial Mult(poly1, poly2) ::= return the polynomial poly1. poly2

2.4.2 Polynomial Representation

- We can represent polynomials by having unique exponents in descending order.
- The following algorithm works by comparing terms from the two polynomials until one or both of the polynomials becomes empty.
- The switch statement performs the comparisons and adds the proper term to the new polynomial d. If one of the polynomials becomes empty, we copy the remaining terms from the non-empty polynomial into d.

Initial version of padd function

```
/* d=a+b, where a,b and d are polynomials */
 d=Zero()
 while(! IsZero(a) && ! IsZero(b)) do {
   switch COMPARE (LeadExp(a), LeadExp(b)) {
      case -1: d=
         Attach(d, Coef(b, LeadExp(b)), LeadExp(b));
         b= Remove(b, LeadExp(b));
         break;
    case 0: sum= Coef(a,LeadExp(a))+Coef(b,LeadExp(b));
       if(sum) {
          Attach(d, sum, LeadExp(a));
          a= Remove(a, LeadExp(a));
          b= Remove(b, LeadExp(b));
       }
       break;
    case 1: d= Attach(d, coef(a,LeadExp(a)), LeadExp(a));
      a=Remove(a, LeadExp(a)); } } insert any remaining terms of a
                                    or b into d.
```