

Polynomial Addition using Link List

```
poly * my_add_poly(poly *phead1, poly *phead2)

p1 = phead1;
p2 = phead2;
poly *p3;
poly *result;
p3 = create new node;
result = p3;
/* now traverse the lists till one list gets exhausted */
while ((p1 != NULL) && (p2 != NULL))
{
    /* if the exponent of p1 is higher than that of p2 then
    the next term in final list is going to be the node of p1* /
    if (p1 ->exp > p2 -> exp )
    {
        p3 -> exp = p1 -> exp;
        p3 -> coff = p1 -> coff ;
        /* now move to the next term in list 1*/
        p1 = p1 -> next;
    }
    /* if p2 exponent turns out to be higher then make p3
    same as p2 and append to final list */
    else if (p1 ->exp < p2 -> exp )
    {
        p3 -> exp = p2 -> exp;
        p3 -> coff = p2 -> coff ;
        p2 = p2 -> next;
    }

    /* both exponents are same, so add the coefficients */
    else if (p1 ->exp = p2 -> exp )
    {
        p3 = allocate new node;
        p3-> exp = p1-> exp;
        p3->coff = p1->coff + p2-> coff ;
        append (p3, phead3) ;
        p1 = p1->next ;
        p2 = p2->next ;
    }
}

If (p1 != null && p2!= null)
    create new node at end of p3;
p3 = p3->next;
} outer while loop
```

```

if ( p1 != NULL)
    while (p1 != null)
    {
        Create new node at end of p3;
        p3 = p3->next;
        p3 -> exp = p1 -> exp;
        p3 -> coff = p1 -> coff ;
        /* now move to the next term in list 1*/
        p1 = p1 -> next;
    }
else if (p2 != null)
    while (p2 != null)
    {
        Create new node at end of p3;
        p3 = p3->next;
        p3 -> exp = p2 -> exp;
        p3 -> coff = p2 -> coff ;
        /* now move to the next term in list 1*/
        P2 = p2 -> next;
    }

return result;

```