

Desarrollar las siguientes actividades:

1. Dado el Polinomio  $P(x) = 3x^6 + 2x^4 + 3$  y el polinomio  $Q(x) = 3x - 1$  cuál es el resultado de aplicar las siguientes expresiones utilizando las Funciones y Axiomas del TAD Polinomio.

$\text{Rem}(P, 4) \rightarrow 3x^6 + 3$

$\text{Rem}(\text{Attach}(P, -8, 5), 0) \rightarrow 3x^6 - 8x^5$

$\text{Iszero}(\text{Attach}(P, 0, 0)) \rightarrow \text{FALSE}$

$\text{Smult}(\text{Zero}, 3, 6) \rightarrow \text{Zero}$

$\text{Smult}(\text{Attach}(P, 2, 2), 1, 1) \rightarrow 3x^7 - 8x^6 + 2x^3$

$\text{Attach}(\text{Smult}(P, 1, 1), 1^*2, 2+1) \rightarrow 3x^8 - 8x^7 + 2x^4 + 2x^3$

$\text{Add}(P, \text{Zero}) \rightarrow 3x^8 - 8x^7 + 2x^4 + 2x^3$

$\text{Add}(P, \text{Attach}(P, 3, 2)) \rightarrow 6x^8 - 16x^7 + 4x^4 + 4x^3 + 3x^2$

$\text{Attach}(\text{Add}(P, Q), 3, 2) \rightarrow 6x^8 - 16x^7 + 4x^4 + 4x^3 + 6x^2 + 3x - 1$

$\text{Mult}(P, \text{Zero}) \rightarrow \text{Zero}$

$\text{Mult}(P, \text{Attach}(Q, 4-2, 3)) \rightarrow \text{Zero}$

$\text{Add}(\text{Mult}(P, Q), \text{Smult}(P, 4-2, 3)) \rightarrow \text{Zero}$

2. Implementar en C# las siguientes operaciones del TAD Polinomios (clsPoly) en el proyecto de clases cApp

Declare Zero() -> Poly //Define polinomio

**Public clsPoly Zero()**

```
{  
    return new clsPoly();  
}
```

Iszero(Poly) -> Boolean //Esta vacío el Polinomio

**Public bool IsZero()**

```
{  
    return (nterm == 0);  
}
```

Coef(Poly, exp) -> coef //Obtiene el coef. del Polinomio

**Public float Coef(int exp)**

```
{  
    return vcoef[exp];  
}
```

Attach(Poly, coef, exp) -> Poly //Adiciona un elemento al Polinomio

**Public clsPoly Attach(int coef, int exp)**

```

{
if ((coef != 0) && (exp >= 0))
{
if (p.Vcoef[exp] == 0)
{
p.nterm++;
}
p.Vexp[exp] = exp;
p.Vcoef[exp] = p.Vcoef[exp] + coef;
}
return p;
}

```

Rem(Poly, exp) -> Poly //Elimina un elemento del Polinomio

**Public clsPoly Rem(int exp)**

```

{
if (exp >= 0)
{
p.Vcoef[exp] = 0;
p.Vexp[exp] = 0;
nterm--;
}
return p;
}

```

Add(Poly, Poly) ->Poly //Adición de Polinomios

**public clsPoly Add(clsPoly P,clsPoly Q)**

```

{

clsPoly C = new clsPoly();

while ((P.IsZero() == true && Q.IsZero() == true) == false)

{

if (P.Grado() < Q.Grado())

{

C = Attach(C, Q.Coef(Q.Grado()), Q.Grado());

Q = Rem(Q, Q.Grado());

}

if (P.Grado() > Q.Grado())

{

```

```

        C = Attach(C, P.Ccoef(P.Grado()), P.Grado());

        P = Rem(P, P.Grado());

    }

    if (P.Grado() == Q.Grado())

    {

        C = Attach(C, P.Ccoef(P.Grado()) + Q.Ccoef(Q.Grado()),
P.Grado());

        P = Rem(P, P.Grado());

        Q = Rem(Q, Q.Grado());

    }

}

return C;

}

```

Grado(Poly) -> N      // Obtiene el grado de un Polinomio

```

public int Grado()

{

    int Exp = 0;

    for (int k = 0; k < MAX; k++)

    {

        if (vExp[k] > 0)

        {

            Exp = vExp[k];

        }

    }

    return Exp;
}

```

}