

## Menú con funciones, no se si lo esta haciendo bien

Escrito por ivanel93 - 10/03/2013 13:16

---

Hola muchachos es la segunda vez que pido ayuda emm tengo un problema? , bueno no he entendido del todo el tema de los nodos y estructuras en c (mi profe no domina el tema, y quiere que le entendamos a la primera), puesto que me han dejado un código incompleto el cual he tratado de terminar bueno principalmente me enfoco a que tengo que desarrollar una rutina de generar la función seno y  $\sinh^{-1}$  pero al ejecutar mi programa no se si estén correctos los valores que este muestra solicito su colaboración para ver si lo que he estado haciendo esta bien, no entiendo el tema espero me ayuden muchas gracias de antemano .

Anexo el código para que lo chequen :

Código: /\* PROGRAMA: SUMA, RESTA, MULTIPLICACION Y DIVISION DE POLINOMIOS \*/

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
#define DELAY 64000
```

```
struct apuntador
```

```
{
```

```
float coef;
```

```
int exp1;
```

```
struct apuntador *siguiente;
```

```
} nodo;
```

```
struct apuntador
```

```
*polydat1, *polydat2, *polysum1, *polysum2, *polysuma, *polymin,
```

```
*polysus, *polyres, *polyfac1, *polyfac2, *polyprod, *polynum,
```

```
*polyden, *polycos, *polyresta, *polyresul, *aux_poly, *nuevo,
```

```
*polyder;
```

```
void pausa(void), crea(void), iniciapoly(void),
```

```
presentacion(void),
```

```
insert_lista(struct apuntador **poly, float c, int e),
```

```
imprime(struct apuntador *poly),
```

```
suma_poly(struct apuntador *poly1, struct apuntador *poly2,
```

```
struct apuntador **poly3),
```

```
multy_poly(struct apuntador *poly1, struct apuntador *poly2,
```

```
struct apuntador **poly3),
```

```
resta_poly(struct apuntador *poly1, struct apuntador *poly2,
```

```
struct apuntador **poly3),
```

```
div_poly(struct apuntador *poly1, struct apuntador *poly2,
```

```
struct apuntador **poly3, struct apuntador **poly4),
```

```
der_poly(struct apuntador *poly1, struct apuntador **poly3),
```

```
borrar_poly(struct apuntador **poly3, int e);
```

```
float eval_poly(struct apuntador *poly1,float x);
float factorial(int);

int seleccion_menu(void);
int i,caso,e;
float c,epsilon;
float coef;
int exp1,fx,x0; // SE CAMBIA EXP SOLO AQUI O EN TODOS?
char n;
int opcion;

main()
{
    presentacion();
    clrscr();
    for(;;)
    {
        switch(seleccion_menu())
        {

            case 1:
                int limite, iter=0;
                float fxd,epsilon,xa,xn;
                clrscr();
                printf("\n Metodo de Newton Raphson (RAICES) de poly1");
                printf("\n Dame el valor inicial de x0   para poly1 ==>");
                scanf("%f",&x0);
                printf("\n Dame el Epsilon ==>");
                scanf("%f",&epsilon);
                printf("\n Dame el nuemro maximo de iteraciones ==>");
                scanf("%d",&limite);
                der_poly(polydat1, &polyder);
                printf("\n #iter----x0-----fx-----fxd-----xn");
                do
                {
                    xa=x0;
                    fx=eval_poly(polydat1,xa);
                    fxd=eval_poly(polyder,xa);
                    xn=xa - fx/fxd;
                    iter=iter+1;
                    printf("\n %d  %f  %f  %f  %f",iter,xa,fx,fxd,xn);
                    x0=xn;
                } while((fabs(xa-xn)>epsilon)&&(iter<limite))
                {
                    //1
                    printf("\n La solucion no esta entre a..b ya que f(a) f(b)>0 n");
                } //1
                else
                while(1)
                {
                    //2
                    it=it+1;
```

```
c=(a+b)/2;
fc=eval_poly(fx,c);
printf("n %3d %10.6f %10.6f %10.6f %10.6f %10.6f",it,a,c,b,fa,fc);
printf("n %14.8f n",fabs((fb-fa)/2));
pausa();
if(it>it_limit)
break;
if(fabs(c-a)>siguiente=NULL;
if(*poly==NULL)
*poly=nuevo;
else
{//-2
aux=*poly;
if(aux->exp1 { //3
nuevo->siguiente=*poly;
*poly=nuevo;
} //-3
else //EN OTROS CASOS
{//4
while((aux)&&(band))
{//5
if(aux->exp1==expo)
{//6 PARA TERMINO CON IGUAL EXPONENTE
aux->coef=aux->coef+coefi;
delete(nuevo); // O ES "if(aux->coef==0)"
band=1;
} //-6
else //PARA EL CASO: aux->exp1>expo
{//7
if((aux->siguiente!= NULL)&&(expo>aux->siguiente->exp1))
{//8
nuevo->siguiente=aux->siguiente;
aux->siguiente=nuevo;
band=1;
} //-8
else
{//9
if(aux->siguiente==NULL)
{//10
aux->siguiente=nuevo;
band=1;
} //-10
else
ant=aux;
aux=aux->siguiente;
} //-9
} //-7
} //-5
} //-4
}
} //-0
```

```
void imprime(struct apuntador *poly)
{
    while(poly)
    {
        printf("\nn Coeficiente : %f n",poly->coef);
        printf(" Exponente   : %i",poly->exp1);
        poly=poly->siguiente;
    }
}

void suma_poly(struct apuntador *poly1, struct apuntador *poly2,
    struct apuntador **poly3)
{
    float suma;
    struct apuntador *primero;
    primero = NULL;
    suma=0;
    while((poly1 != NULL) && (poly2 != NULL))
    {
        if (poly1->exp1 == poly2->exp1)
        {
            suma=poly1->coef + poly2->coef;
            if(suma !=0)
            {
                insert_lista(&primero,suma,poly1->exp1);
            }
            poly1=poly1->siguiente;
            poly2=poly2->siguiente;
        }
        else
        {
            if(poly1->exp1 > poly2->exp1)
            {
                insert_lista(&primero,poly1->coef,poly1->exp1);
                poly1=poly1->siguiente;
            }
            else
            {
                insert_lista(&primero,poly2->coef,poly2->exp1);
                poly2=poly2->siguiente;
            }
            suma=0;
        }
    }
    while(poly1 != NULL)
    {
        insert_lista(&primero,poly1->coef,poly1->exp1);
        poly1=poly1->siguiente;
    }
    while(poly2 != NULL)
    {
        insert_lista(&primero,poly2->coef,poly2->exp1);
        poly2=poly2->siguiente;
    }
}
```

```
*poly3 = primero;
}

void multy_poly(struct apuntador *poly1, struct apuntador *poly2,
    struct apuntador **poly3)
{
    struct apuntador *primero;
    float multy;
    int sum;
    struct apuntador *p, *q, *aux1, *aux2;

    primero = NULL;
    *poly3=NULL;
    multy=1;
    sum=0;
    while(poly2 != NULL)
    {
        p=poly1;
        while(p!=NULL)
        {
            multy=p->coef *poly2->coef;
            sum=p->exp1 + poly2->exp1;
            if(multy != 0)
            {
                insert_lista(&primero,multy,sum);
                p=p->siguiente;
            }
            else
            {
                insert_lista(&primero,p->coef,p->exp1);
                p=p->siguiente;
                poly2=poly2->siguiente;
            }
            multy=1;
        }
        poly2=poly2->siguiente;
    }
    q = primero;
    while(q)
    {
        if(q->exp1==q->siguiente->exp1)
        {
            q->coef=q->coef + q->siguiente->coef;
            q->siguiente=q->siguiente->siguiente;
        }
        else
            q=q->siguiente;
    }
    *poly3=primero;
}

void resta_poly(struct apuntador *poly1, struct apuntador *poly2,
```

```
struct apuntador **poly3)
{
    float resta;
    struct apuntador *primero;
    primero = NULL;
    resta= 0;
    while ((poly1 != NULL) && (poly2 != NULL))
    {
        if (poly1->exp1 == poly2->exp1)
        {
            resta= poly1->coef - poly2->coef;
            if (resta != 0) {
                insert_lista (&primero, resta, poly1->exp1); }
            poly1= poly1->siguiente;
            poly2= poly2->siguiente;
        }
        else
            if (poly1->exp1 > poly2->exp1)
            {
                insert_lista (&primero, poly1->coef, poly1->exp1);
                poly1= poly1->siguiente;
            }
            else
            {
                insert_lista (&primero, poly2->coef, poly2->exp1);
                poly2= poly2->siguiente;
            }
    }
    resta= 0;
}

while (poly1 != NULL)
{
    insert_lista (&primero, poly1->coef, poly1->exp1);
    poly1= poly1->siguiente;
}

while (poly2 != NULL)
{
    insert_lista (&primero, -poly2->coef, poly2->exp1);
    poly2= poly2->siguiente;
}
*poly3 = primero;
}

void borrar_poly(struct apuntador **poly, int expo)
{
    // aqui va el codigo correspondiente
}

void div_poly(struct apuntador *polynume, struct apuntador *polydeno,
    struct apuntador **polycoci, struct apuntador **polyresi)
{
```

```
}

void der_poly(struct apuntador *poly1,struct apuntador **poly3)
{
    struct apuntador *primero=NULL;
    float co=0;
    int ex=0;
    while((poly1!=NULL)&&(poly1->exp1!=0))
    {
        co=poly1->coef*poly1->exp1;
        ex=poly1->exp1-1;
        insert_lista(&primero,co,ex);
        poly1=poly1->siguiente;
    }
}

float eval_poly(struct apuntador *poly1, float ev)
{
    float res2;
    res2=0;
    while(poly1)
    {
        res2=res2+poly1->coef*pow(ev,poly1->exp1);
        poly1=poly1->siguiente;
    }
    return (res2);
}

void presentacion(void)
{
    char c;
    clrscr();
    printf("\nn          PROGRAMA POLINOMIO nn");
    printf("\n          REALIZA LA SUMA, RESTA, MULTIPLICACION Y DIVISION nn");

}

float factorial(int n)
{
    int k;
    float f=1;
    for(k=1;k
```

=====

## Re: Menú con funciones, no se si lo esta haciendo bien

Escrito por adrianvaca - 11/03/2013 11:13

---

Qué compilador estás usando?

Puedes compilar y ejecutar el programa para probar si este funciona bien, a la par hacer usar la

calculadora para verificar si los resultados son correctos o no.

=====