



Centro de Enseñanza Técnica Industrial

Plantel Colomos

Ingeniería en Desarrollo de Software

Nombre Alumno: José Rafael Ruiz Gudiño

Registro: 20110374

Arquitectura de Computadoras

Práctica 2: Suma / resta en punto flotante

3°P

T/M

08/03/2021

## Captura del programa

### Suma

```
C:\Users\Usuario\Documents\Progra-101\3 Semestre\Arquitectura de computadoras\SinNombre1.exe
1-Suma
2-Resta
3-Salir.
Escoge una opcion: 1
Ingresa un numero: 43.75
Ingresa otro numero: .25

43.750000 n1
0.250000 n2

Numero 1 en binario:
101011.11
Numero 2 en binario:
.01
Binario 1 en exponencial:
1.0101111x2^5
Binario 2 en exponencial:
1.x2^-2
Suma:

01010111100000000000000000000000 +
00000000100000000000000000000000
-----
01011000000000000000000000000000
Resultado suma:
1.011000000000000000000000x2^5
Resultado suma en Decimal: 44.000000
-----
Process exited after 15.74 seconds with return value 0
Presione una tecla para continuar . . .
```

### Resta

```
Selecccionar C:\Users\Usuario\Documents\Progra-101\3 Semestre\Arquitectura de computadoras\SinNombre1.exe
1-Suma
2-Resta
3-Salir.
Escoge una opcion: 2
Ingresa un numero: 43.75
Ingresa otro numero: .25

Numero 1 en binario:
101011.11
Numero 2 en binario:
.01
Binario 1 en exponencial:
1.0101111x2^5
Binario 2 en exponencial:
1.x2^-2
Resta:

01010111100000000000000000000000 -
00000000100000000000000000000000
-----
01010111000000000000000000000000
Resultado resta:
1.010110000000000000000000x2^5
Resultado resta en Decimal: 43.500000
-----
Process exited after 10.32 seconds with return value 0
Presione una tecla para continuar . . .
```

## Código del Programa

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
#include<math.h>

int main(){
    char
    binario1[100],binariof1[100],binariofull1[100],binariofull2[100],resum[100],rerest[100],resul1[24],resul2[24]; //cadenas para almacenar el primer numero a sumar/restar

    char
    binario2[100],binariof2[100],bint[24],bif[24]; //cadenas para almacenar el segundo numero a sumar/restar

    int
    exp1,longf1,exp2,longf2,cont,temp,long1,long2,difexp,sumdec,sumint=0,sumf=0;
    //para determinar el exponente y longitud de las cadenas

    int
    i=0,j=0,numint1,numint2,carry,sum,rest,aux,cont2=0,opc,cero=0; //contadores y variable para obtener parte entera

    float numf1,numf2,n1,n2,aux3;
    //Para leer los numeros numero

    double dcimal1,dcimal2; //Para obtener la parte fraccionaria

    printf("\n1-Suma\n2-Resta\n3-Salir.\nEscoge una opcion: ");
    scanf("%d",&opc);

    switch(opc){
        case 1:
            printf("Ingresa un numero: ");
            scanf("%f",&numf1);

            printf("Ingresa otro numero: ");
            scanf("%f",&numf2);

            if(numf1<0){ numf1*=-1;} //por si es negativo

            if(numf2<0){numf2*=-1;}

            if(numf1<numf2){
                aux3=numf1;
                numf1=numf2;
                numf2=aux3;
            }

            printf("\n%f n1",numf1);
            printf("\n%f n2\n",numf2);

            n1=numf1;
            n2=numf2;

            numint1= numf1;    //Se obtiene la parte entera

            dcimal1 = (numf1)-numint1;

            numint2= numf2;

            dcimal2 = (numf2)-numint2;

            //CONVERSION DEL SISTEMA DECIMAL A BINARIO

            //PARTE ENTERA

            //Primer numero

            while(numint1>0){    // 5>0// 2>0//
                1>0//
```

```

        binario1[i]=
numint1%2;//bin[0] = 1 | bin[1]= 0| bin[2]=
1
        numint1/=2; //num=2 //
num=1| num=0
        ++i; //i=1|i=2|i=3
    } exp1=i;// exponente
//PARTE FRACCIONARIA
while(dcimal1!=0){
    dcimal1*=2;
    if(dcimal1<1){
        binariof1[j]= 0;
        j++;
    } else{
        binariof1[j]=1;
        dcimal1-=1;
        j++;
    }
} longf1=j;

//Segundo numero
i=0;j=0; //Se vuelven a igualar a 0
los contadores para utilizarlos
while(numint2>0){
    binario2[i]= numint2%2;
    numint2/=2;
    ++i;
} exp2=i;// exponente
//PARTE FRACCIONARIA
while(dcimal2!=0){
    dcimal2*=2;

```

```

        if(dcimal2<1){
            binariof2[j]= 0;
            j++;
        } else{
            binariof2[j]=1;
            dcimal2-=1;
            j++;
        }
    } longf2=j;

    //Se invierte el orden de las cadenas
de la parte entera
    cont=0; temp=0;
    while(cont < exp1/2) //invierte el orden
    {
        temp = binario1[cont];
        binario1[cont] = binario1[exp1 - 1 -
cont];
        binario1[exp1 - 1 - cont] = temp;
        cont++;
    }
    cont=0; temp=0;
    while(cont < exp2/2) //invierte el orden
    {
        temp = binario2[cont];
        binario2[cont] = binario2[exp2 - 1 -
cont];
        binario2[exp2 - 1 - cont] = temp;
        cont++;
    }

```

```

//IMPRESION
//Numero 1
printf("\nNumero 1 en binario: \n");

for(i=0;i<exp1;++i){ //Entero

    printf("%d",binario1[i]);
}
printf(".");
for(j=0;j<longf1;j++){ //fraccion

    printf("%d",binariof1[j]);
}

//Numero 2
printf("\nNumero 2 en binario: \n");

for(i=0;i<exp2;++i){ //Entero

    printf("%d",binario2[i]);
}
printf(".");
for(j=0;j<longf2;j++){ //fraccion

    printf("%d",binariof2[j]);
}

////////////////////
PASAR A MODO EXPONENCIAL

    printf("\nBinario 1 en
exponencial:\n"); //NUMERO BINARIO
EN EXPONENCIAL

```

```

for(i=0;i<exp1;++i){

    printf("%d",binario1[i]);
    if(i==0) printf(".");
}

if(exp1==0){
    i=0;
    while(binariof1[i]!=1){
        if(binariof1[i]==0){
            cero++;
            i++;
        }
    }

    for(i=cero;i<longf1;i++){
        printf("%d",binariof1[i]);
        if(i==cero&&zero<=1)
            printf("x2^-%d",cero+1);
        } else{
            for(j=0;j<longf1;j++){

                printf("%d",binariof1[j]);

            }printf("x2^%d",exp1-1); }

//NUMERO 2

```

```

printf("\nBinario 2 en
exponencial:\n"); //NUMERO BINARIO
EN EXPONENCIAL

```

```

for(i=0;i<exp2;++i){

```

```

    printf("%d",binario2[i]);

```

```

    if(i==0) printf(".");

```

```

}

```

```

if(exp2==0){

```

```

    i=0;

```

```

    while(binario2[i]!=1){

```

```

        if(binario2[i]==0){

```

```

            cero++;

```

```

            i++;

```

```

        }

```

```

    }

```

```

        for(i=cero;i<longf2;i++){

```

```

            printf("%d",binario2[i]);

```

```

            if(i==cero&&zero<=1)

```

```

printf(".");

```

```

        }printf("x2^-%d",zero+1);

```

```

    } else{

```

```

        for(j=0;j<longf2;j++){

```

```

            printf("%d",binario2[j]);

```

```

            if(exp2==0&&j==0&&longf2>1)

```

```

printf("."); //por si es 0 el entero

```

```

        }printf("x2^-%d",exp2-1);}

```

```

        if(n1>=n2){

```

```

//concatenar las cadenas

```

```

printf("\nSuma:\n");

```

```

//cadena 1

```

```

/////

```

```

printf("\n");

```

```

long1 = exp1+longf1;

```

```

difexp = exp1-exp2;

```

```

// printf("\ndifexp: %d\n",difexp);

```

```

for(i=1;i<exp1+1;i++){

```

```

    binariofull1[i]=binario1[i-1];

```

```

}

```

```

for(i=exp1+1;i<long1+1;i++){

```

```

    binariofull1[i]=binario1[i-exp1-1];

```

```

}

```

```

for(i=long1+1;i<33;i++){

```

```

    binariofull1[i]=0;

```

```

}

```

```

binariofull1[0]=0;

```

```

for(i=0;i<33;i++){

```

```

    printf("%d",binariofull1[i]);

```

```

} printf(" +");

```

```

//cadena 2

```

```

long2=exp2+longf2;

```

```

printf("\n");

```

```

for(i=1;i<difexp+1;i++){

```

```

    binariofull2[i]=0;

```

```

}

```

```

        for(i=difexp+1;i<exp2+difexp+1;i+
+){
            binariofull2[i]=binario2[i-difexp-1];
        }
        for(i=exp2+difexp+1;i<long2+difex
p+1;i++){
            binariofull2[i]=binario2[i-
exp2-difexp-1];
        }
        for(i=long2+difexp+1;i<33;i++){
            binariofull2[i] = 0;
        }
        binariofull2[0]=0;

        for(i=0;i<33;i++){
            printf("%d",binariofull2[i]);
        }
        printf("\n");
        for(i=0;i<33;i++){
            printf("-");
        }
        i=exp1;
        //SUMA
        carry=0;
        for(i=32;i>=0;i--){

            sum=carry+binariofull1[i]+binariof
ull2[i];

            switch(sum){
                case 0: resum[i]=0;
                carry=0; break;

                case 1: resum[i]=1;
                carry=0; break;

```

```

                case 2: resum[i]=0;
                carry=1; break;

                case 3: resum[i]=1;
                carry=1; break;
            }}
            printf("\n");
            for(i=0;i<32;i++){
                printf("%d",resum[i]);
            }
            //Se toman los 24 bits de
            resultado
            for(i=0;i<24;i++){ //Suma
                resul1[i]=resum[i];
            }
            aux=exp1;
            if(resul1[0]==1) aux++; //por si el
            primer numero es 1 en la suma
            if(resul1[0]==0){ //elimina el cero a
            la izquierda en la suma (si la hay)
                for(i=1;i<24;i++){
                    resul1[i-1]=resul1[i];
                }
            }
            //imprmir el resultado
            printf("\nResultado suma:\n");
            for(i=0;i<24;i++){ //Suma
                printf("%d",resul1[i]);
                if(i==0) printf(".");
            } printf("x2^%d",aux-1);
            ///Conversion binario a decimal
            for(i=0;i<exp1;i++){
                bint[i]=resul1[i];
            }

```

```

//      printf("\nParte entera suma: ");
//      for(i=0;i<aux-1;i++){
//          printf("%d",bint[i]);
//      }
//      printf("\nParte fraccionaria suma:
//      .");
//          for(i=exp1;i<24;i++){
//              bif[i-exp1]=resul1[i];
//          }
//      for(i=0;i<24-exp1;i++){
//          printf("%d",bif[i]);
//      }
//      j=aux-1;
//      sumint=0;

//          for(i=0;i<aux;i++){
//              if(bint[i]==1){cont2=
//              pow(2,i);} else cont2=0;
//              sumint+=cont2;
//              j--;
//          }
//          printf("\nResultado suma en
//          Decimal: %f ",n1+n2);
//          fflush(stdin);

//              break;
//          case 2:

//              printf("Ingresa un numero: ");
//              scanf("%f",&numf1);

//              printf("Ingresa otro numero:
//              "); scanf("%f",&numf2);

//              if(numf1<0){ numf1*=-1;}
//              //por si es negativo

```

```

if(numf2<0){numf2*=-1;}

if(numf1<numf2){
    aux3=numf1;
    numf1=numf2;
    numf2=aux3;
}

n1=numf1;
n2=numf2;

if(numf1<0) numf1*=-1; //por si es
negativo

if(numf2<0) numf2*=-1;

numint1= numf1;    //Se obtiene
la parte entera

dcimal1 = (numf1)-numint1;

numint2= numf2;
dcimal2 = (numf2)-numint2;

//CONVERSION DEL SISTEMA
DECIMAL A BINARIO

//PARTE ENTERA

//Primer numero

while(numint1>0){    // 5>0// 2>0//
1>0//

    binario1[i]=
numint1%2;//bin[0] = 1 | bin[1]= 0| bin[2]=
1

    numint1/=2; //num=2    //
num=1| num=0

    ++i; //i=1|i=2|i=3
} exp1=i;// exponente

//PARTE FRACCIONARIA

while(dcimal1!=0){

```



```

dcimal1*=2;
if(dcimal1<1){
    binariof1[j]= 0;
    j++;
} else{
    binariof1[j]=1;
    dcimal1-=1;
    j++;
}
} longf1=j;

//Segundo numero
i=0;j=0; //Se vuelven a igualar a 0
los contadores para utilizarlos
while(numint2>0){
    binario2[i]= numint2%2;
    numint2/=2;
    ++i;
} exp2=i;// exponente
//PARTE FRACCIONARIA
while(dcimal2!=0){
    dcimal2*=2;
    if(dcimal2<1){
        binariof2[j]= 0;
        j++;
    } else{
        binariof2[j]=1;
        dcimal2-=1;
        j++;
    }
}

} longf2=j;

//Se invierte el orden de las cadenas
de la parte entera
cont=0; temp=0;
while(cont < exp1/2) //invierte el orden
{
    temp = binario1[cont];
    binario1[cont] = binario1[exp1 - 1 -
cont];
    binario1[exp1 - 1 - cont] = temp;
    cont++;
}
cont=0; temp=0;
while(cont < exp2/2) //invierte el orden
{
    temp = binario2[cont];
    binario2[cont] = binario2[exp2 - 1 -
cont];
    binario2[exp2 - 1 - cont] = temp;
    cont++;
}

//IMPRESION
//Numero 1
printf("\nNumero 1 en binario: \n");

for(i=0;i<exp1;++i){ //Entero

    printf("%d",binario1[i]);

}

```

```

printf(".");
for(j=0;j<longf1;j++){ //fraccion

printf("%d",binariof1[j]);
}

//Numero 2
printf("\nNumero 2 en binario: \n");

for(i=0;i<exp2;++i){ //Entero

printf("%d",binario2[i]);
}

printf(".");
for(j=0;j<longf2;j++){ //fraccion

printf("%d",binariof2[j]);
}

////////////////////
PASAR A MODO EXPONENCIAL

printf("\nBinario 1 en
exponencial:\n"); //NUMERO BINARIO
EN EXPONENCIAL

for(i=0;i<exp1;++i){

printf("%d",binario1[i]);
if(i==0) printf(".");
}

if(exp1==0){
i=0;
while(binariof1[i]!=1){

printf(".");
if(binariof1[i]==0){
cero++;
i++;
}

}

for(i=cero;i<longf1;i++){
printf("%d",binariof1[i]);
if(i==cero&&zero<=1)
printf("x2^-%d",cero+1);
cero=0;
} else{
for(j=0;j<longf1;j++){
printf("%d",binariof1[j]);
}printf("x2^%d",exp1-1);}

//NUMERO 2

printf("\nBinario 2 en
exponencial:\n"); //NUMERO BINARIO
EN EXPONENCIAL

for(i=0;i<exp2;++i){

printf("%d",binario2[i]);
if(i==0) printf(".");
}

if(exp2==0){
i=0;
while(binariof2[i]!=1){
if(binariof2[i]==0){
cero++;

```

```

        i++;
    }
}

for(i=cero;i<longf2;i++){
    printf("%d",binariof2[i]);
    if(i==cero&&zero<=1)
printf(".");
    }printf("x2^-%d",zero+1);
zero=0;
    } else{
        //if(exp2==0) printf("."); exp2=1;
        ///////POR SI el exponenete es 0
        for(j=0;j<longf2;j++){

            printf("%d",binariof2[j]);

            if(exp2==0&&j==0&&longf2>1)
printf(".");

        }

        printf("x2^-%d",exp2-1); }

        if(n1>=n2){
///concatenar las cadenas
//cadena 1
/////
long1 = exp1+longf1;
difexp = exp1-exp2;
// printf("\ndifexp: %d\n",difexp);
for(i=1;i<exp1+1;i++){
    binariofull1[i]=binario1[i-1];
}
for(i=exp1+1;i<long1+1;i++){

```

```

    binariofull1[i]=binariof1[i-exp1-1];
}
for(i=long1+1;i<33;i++){
    binariofull1[i]=0;
}
    binariofull1[0]=0;

//cadena 2
    long2=exp2+longf2;

    for(i=1;i<difexp+1;i++){
        binariofull2[i]=0;
    }
    for(i=difexp+1;i<exp2+difexp+1;i+
+){
        binariofull2[i]=binario2[i-difexp-1];
    }
    for(i=exp2+difexp+1;i<long2+difexp+1;i++){
        binariofull2[i]=binariof2[i-
exp2-difexp-1];
    }
    for(i=long2+difexp+1;i<33;i++){
        binariofull2[i] = 0;
    }
    binariofull2[0]=0;
    i=exp1;

//Resta
    printf("\n\nResta:\n");
    for(i=0;i<33;i++){

```

```

printf("%d",binariofull1[i]);
} printf(" -\n");
    for(i=0;i<33;i++){
        printf("%d",binariofull2[i]);
    }
printf("\n");
for(i=0;i<33;i++){
    printf("-");
}
carry=0;
for(i=32;i>=0;i--){
    rest=binariofull1[i]-carry-
binariofull2[i];
    switch(rest){
        case 0:
if(binariofull1[i]==1&&carry==1&&binariof
ull2[i]==0){
            rerest[i]=0;
            carry=0; } else
if(binariofull1[i]==1&&carry==0&&binariof
ull2[i]==1){
            rerest[i]=0;
            carry=0;} else
if(binariofull1[i]==0&&carry==0&&binariof
ull2[i]==0){
            rerest[i]=0;
            carry=0;}

        break;

        case 1:
if(binariofull1[i]==1&&carry==0&&binariof
ull2[i]==0){

rerest[i]=1; carry=0;}

        break;

```

```

        case -1:
if(binariofull1[i]==1&&carry==1&&binariof
ull2[i]==1){

rerest[i]=1; carry=1;} else
if(binariofull1[i]==0&&carry==1&&binariof
ull2[i]==0){

rerest[i]=1; carry=1;} else
if(binariofull1[i]==0&&carry==0&&binariof
ull2[i]==1){

rerest[i]=1; carry=1;}

        break;

        case -2:
if(binariofull1[i]==0&&carry==1&&binariof
ull2[i]==1){

rerest[i]=0; carry=1;}

        break;

    }
}
printf("\n");
for(i=0;i<32;i++){
    printf("%d",rerest[i]);
}

for(i=0;i<24;i++){ //Resta
    resul2[i]=rerest[i];
}

if(resul2[0]==0){ //elimina el cero a la
izquierda en la resta (si la hay)

    for(i=1;i<24;i++){

        resul2[i-1]=resul2[i];

    }
}

```

```

}printf("\nResultado resta:\n");
for(i=0;i<24;i++){ //Resta
    printf("%d",resul2[i]);
    if(i==0) printf(".");
} printf("x2^%d",exp1-1);}
///Conversion binario a
decimal
for(i=0;i<exp1;i++){
    bint[i]=resul1[i];
}

// printf("\nParte entera resta: ");
// for(i=0;i<aux-1;i++){
//     printf("%d",bint[i]);
// }
// printf("\nParte fraccionaria resta:
// .");
for(i=exp1;i<24;i++){
    biff[i-exp1]=resul1[i];
}

// for(i=0;i<24-exp1;i++){
//     printf("%d",biff[i]);
// }
j=aux-1;
sumint=0;
for(i=0;i<aux;i++){
    if(bint[i]==1){cont2=
    pow(2,i);} else cont2=0;
    sumint+=cont2;
    j--;
}
printf("\nResultado resta en
Decimal: %f ",n1-n2);
break;
default: break;
}}
return 0;}

```

## Conclusión

Este programa estaba más complicada que la anterior ya que se tornaba más complicado el hecho de tomar en cuenta las posiciones de las cadenas para de esta manera sumarlos o restarlos según el caso. En mi caso lo más complicado fue la operación de realizar la suma y resta debido a que se tenía que tener mucho cuidado con el acarreo en las operaciones, pero en mi punto de vista fue más complicada realizar la resta ya que tuve que realizar una tabla de verdad para todos los casos posibles en la resta a través de un switch. Aunque en esta práctica estuvo más complicada me gustó y pareció interesante ya que tuve que pensar mucho más. Aparte de que aprendí a sumar y restar en binario y saber lo que se tiene que tomar en cuenta.