

Implementación del Algoritmo J en C

Ing. Vicente Oscar Mier Vela

8 de julio de 2013

El siguiente programa en C recibe un arreglo de valores numéricos para asignarle valores constantes a cada una de las variables de la siguiente sumatoria:

$$\sum_{i=1}^n A_1 i^1 + A_2 i^2 + \dots + A_\delta i^\delta$$

En el programa, el primer argumento representa el valor de la variable n , y los siguientes argumentos representan, en orden, los valores de los coeficientes $A_1, A_2, \dots, A_\delta$

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

double j_alg(int, double, double);

int main(int argc, char *argv[]) {
    int i = 0;
    int num = atoi(argv[1]);
    double power, coef, total;
    power = coef = total = 0;
    for(i=2; i<argc; i++){
        coef = atof(argv[i]);
        power = i-1;
        total+=j_alg(power, coef, num);
        printf("\n");
    }
    printf("--\n\nPolinomio en i:\n\t");
    for(i=2; i<argc; i++){
        coef = atof(argv[i]);
        power = i-1;
        if(coef>0){
            printf("+%.2f*i^%.2f", coef, power);
        }
        if(coef<0){
            printf("%.2f*i^%.2f", coef, power);
        }
    }
}
```

```

    }
}
printf("\nResultado de la sumatoria \
para n=%d:\n\t%.2f\n",num,total);
return 0;
}

double j_alg(int power, double coef, double num) {
printf("%.2f*i^%d\n",coef,power);
power+=1;
printf("n = %.2f\n",num);
int i, j, k;
i = j = k = 0;
double mat[power][power+1];
double coefs[power];
for(i=0;i<power;i++){
    for(j=0;j<power+1;j++){
        mat[i][j]=0;
    }
}
for(i=0;i<power;i++){
    mat[0][i]=1;
}
for(i=1;i<power;i++){
    mat[i][i]=i+1;
}
for(i=1;i<power;i++){
    for(j=i+1;j<power;j++){
        mat[i][j]=mat[i-1][j-1]+mat[i][j-1];
    }
}
for(i=0;i<power;i++){
    for(j=i;j<power;j++){
        if(((i+j)%2)>0){
            mat[i][j]*=-1;
        }
    }
}
mat[power-1][power] = coef;
for(i=0;i<power;i++){
    for(j=0;j<power+1;j++){
        printf("%.2f\t",mat[i][j]);
    }
printf("\n");
}
double a,b;

```

```

a = mat[power-1][power];
b = mat[power-1][power-1];
coefs[power-1] = a/b;
for(i=power-2;i>=0;i--){
    double sum = 0;
    for(j=power-1;j>i;j--){
        sum+=mat[i][j]*coefs[j];
    }
    a = -1*sum;
    b = mat[i][i];
    coefs[i]=a/b;
}
for(i=0;i<power;i++){
    printf("B%d\t=\t%f\n",i+1,coefs[i]);
}
double suma = 0;
for(i=0;i<power;i++){
    if(coefs[i]>0){
        suma += num*pow(coefs[i],i+1);
    }
}
printf("Sumatoria para n=%.2f: %.2f\n",num,suma);
return suma;
}

```

El programa se compila usando gcc (Debian 4.7.2-5) 4.7.2 de la siguiente manera:

```
gcc -o j_alg j_alg.c -lm
```

Al ejecutar el comando

```
./j_alg 3 1 2 -2
```

se obtiene el resultado de la siguiente sumatoria:

$$\sum_{i=1}^3 i + 2i^2 - 2i^3$$

El programa arroja la siguiente salida en stdout:

```

1.00*i^1
n = 3.00
1.00    -1.00    0.00
0.00    2.00    1.00
B1      =        0.500000
B2      =        0.500000
Sumatoria para n=3.00: 2.25

```

```

2.00*i^2
n = 3.00
1.00      -1.00      1.00      0.00
0.00      2.00      -3.00      0.00
0.00      0.00      3.00      2.00
B1        =          0.333333
B2        =          1.000000
B3        =          0.666667
Sumatoria para n=3.00: 4.89

-2.00*i^3
n = 3.00
1.00      -1.00      1.00      -1.00      0.00
0.00      2.00      -3.00      4.00      0.00
0.00      0.00      3.00      -6.00      0.00
0.00      0.00      0.00      4.00      -2.00
B1        =          -0.000000
B2        =          -0.500000
B3        =          -1.000000
B4        =          -0.500000
Sumatoria para n=3.00: 0.00

--

Polinomio en i:
      +1.00*i^1.00+2.00*i^2.00-2.00*i^3.00
Resultado de la sumatoria para n=3:
      7.14

```