Universidad Simón Bolívar 12/10/18

CI2126: Computación 2

Michelle Hernández 15-10694

Marco Dávila 15-10356

**Práctica de Laboratorio #2**

1. **Mis primeros programas Recursivos**

1.a) Escriba un programa que permita chequear los códigos del factorial recursivo y no recursivo

#include <stdio.h>

int facR (int n);

int fac (int n);

int main()

{

int n, recursivo, no\_recursivo;

printf("\n Introduzca e valor al que le desea calcular el factorial:");

scanf("%d", &n);

recursivo= facR(n); //llamado a la funcion recursiva

no\_recursivo= fac(n); // llamano a la funcion no recursiva

if (recursivo== no\_recursivo) printf(" \n El factorial de %d es: %d", n, recursivo); //verifica que son iguales

return 0;

}

int facR(int n){

if (n == 0) return 1; else return n\*facR(n-1);

}

int fac(int n){

int r = 1, k = 0;

while (k<n) { r = r\*(k+1); k++;}

return r;

}

1.b) Máximo Común Divisor. Escriba dos funciones: una iterativa y una recursiva que permitan calcular el máximo común divisor positivo de dos enteros no ambos nulos a, b. Sus nombres deben ser mcd y mcdR respectivamente.

int mcdR(int a, int b){

if (b == 0) return a; else return mcdR(b, a%b);

}

int mcd(int a, int b){

int k=1;

while (k!=0) { k=a%b; a=b; b=k;}

return a;

}

* 1. Escriba un programa que lea un par de enteros a, b, con b no nulo que representan una fracción, y dé como salida la fracción irreducible correspondiente.

#include <stdio.h>

int mcdR(int a, int b){

if (b == 0) return a; else return mcdR(b, a%b);

}

int main(){

int a, b, mcd;

printf("Introduzca el numerador y el denominador");

scanf("%d %d", &a, &b);

printf("\n %d/%d equivale a: ", a, b);

mcd = mcdR(a,b);

a=a/mcd;

b=b/mcd;

printf("\n %d/%d", a, b);

return 0;

}

1.2) Use que mcd(a, b) mcm(a, b) = ab para escribir una función que permita calcular el mínimo

común múltiplo entre dos enteros no negativos a, b no ambos nulos.

#include <stdio.h>

int mcdR(int a, int b){

if (b == 0) return a; else return mcdR(b, a%b);

}

int main(){

int a, b;

printf("Introduzca a y b");

scanf("%d %d", &a, &b);

printf("El mcm de a y b es: %d", a\*b/mcdR(a, b));

return 0;

}

1. **Búsqueda Binaria Recursiva.**

2.a) Haga un NUEVO programa que permita probar el algoritmo de búsqueda binaria recursivo visto en clase: El algoritmo esta vez debe tener como interfaz: void bbR(int E, int x, int y, int a[]) donde [x, y) representa el intervalo cerrado-abierto de búsqueda, y E el elemento que se quiere buscar en el arreglo ordenado no decrecientemente a.

#include<stdio.h>

int bb(int E, int n, int a[]){

int x = 0, y = n, m;

while (x+1 != y){

m = (x+y)/2;

if (E < a[m]) y = m; else x = m;

}

return a[x] == E;

}

void leerArreglo(int a[],int n){

int k=0;

printf("\nCompletamos el arreglo");

while(k<n){

printf("\nIngrese el valor de la posicion A[%d]=",k);

scanf("%d",&a[k]);

k++;

}

}

void escribirArreglo(int a[],int n){

int k=0;

printf("\n[");

while(k<n)printf("%d ",a[k++]);

printf("]\n");

}

void bbR(int E, int x, int y, int a[]){

if(x+1==y){

if (a[x] == E) printf("SI");

else printf("NO");

}

else {

int n;

n=(x+y)/2;

if(E>=a[n])x=n;

else y=n;

bbR(E, x, y, a);

};

}

int main(){

int n=8,b;

int a[n];

leerArreglo(a,n);

escribirArreglo(a,n);

printf("\nIntroduzca el numero que busca \n");

scanf("%d",&b);

printf ("\n Segun MI programa, el numero ");

bbR(b, 0, n, a);

printf(" se encuentra en el arreglo");

if (bb(b, n, a)) printf ("\n Segun el programa iterativo, el numero SI esta en el arreglo");

else printf ("\n Segun el programa iterativo, el numero No esta en el arreglo");

return 0;

}

2.b) Modifique el algoritmo de búsqueda binaria recursiva que obtuvo antes para que devuelva el lugar donde lo encontró o n si no lo encontró.

int posbbR(int E, int x, int y, int a[]){

if(x+1==y){

if(a[x] == E)return x;

else return 0;

}

else {

int n;

n=(x+y)/2;

if(E>=a[n])x=n;

else y=n;

return posbbR(E, x, y, a);

};

}

1. **El método de Bisección:** Use el esquema de búsqueda binaria para implementar este método completando el esqueleto.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

float f(float x){

return x\*x - 2; //Función x al cuadrado menos 2

}

///Precondición: f(a)\*f(b) < 0, a < b, er > 0 cercano a cero

float biseccion( float a, float b, float er){

float m=(b+a)/2;

while (fabs(f(m))>er ){

if(f(a)> f(b)){

if(f(a)\*f(m)<0) b=m;

else if (f(a)\*f(m)>0) a=m;

}

else if(f(a)<f(b)){

if(f(b)\*f(m)>0) b=m;

else if (f(b)\*f(m)<0) a=m;

}

else if (f(a)==f(b)) break;

m = (b+a)/2;

printf("%f \n", m);

}

return m;

}

int main(){

float a = 1.0, b = 2.0; //Intervalo de busqueda

float r = biseccion(a,b,.000001);

printf("\nPrueba del Metodo de Biseccion:\n");

printf("\n%9.6f es una raiz de la FUNCION f.",r);

return 0;

}