

Control N°1 - ICI 142
FUNDAMENTOS DE PROGRAMACIÓN
Profesores: Wenceslao Palma, Laura Griffiths
Ayudantes: Bertty Contreras, Marcela Fernández
Jorge Araya, Enzo Barbaguelatta
Fecha: 06 Septiembre de 2012

▪ **Pregunta 1: (35 puntos)**

```
#include <stdio.h>
void metodo(int n, int mat[n][n],int vec[n]){
    float vector_x[n], respaldo[n], sumatoria;
    int i,j,iteracion;
    for(i = 0; i < n; i++) vector_x[i] = 0;
    for(iteracion = 1; iteracion <= 1500; iteracion++){
        for(i = 0; i < n; i++){
            sumatoria=0;
            for(j = 0; j < n; j++){
                if(i != j){
                    sumatoria += (float) mat[i][j]*vector_x[j];
                }
            }
            respaldo[i] = (float)((float)vec[i] - (float)sumatoria) / (float)mat[i][i];
        }
        for(i = 0; i < n; i++) vector_x[i]=respaldo[i];
    }
    for(i = 0; i < n; i++) printf("X(%d) = %f\n",i,vector_x[i]);
    return;
}
int main(){
    /** int n = 3;
    int matriz[n][n];
    int vec[n];
    matriz[0][0] = 1;matriz[0][1] = 2;matriz[0][2] = 3;
    matriz[1][0] = 4;matriz[1][1] = 5;matriz[1][2] = 6;
    matriz[2][0] = 7;matriz[2][1] = 8;matriz[2][2] = 9;
    vec[0] = 11;vec[1] = 12;vec[2] = 13;*/
    int n = 2;
    int matriz[n][n];
    int vec[n];
    matriz[0][0] = 2;matriz[0][1] = 1;
    matriz[1][0] = 5;matriz[1][1] = 7;
    vec[0] = 11;vec[1] = 13;
    metodo(n,matriz,vec);
    return 0;
}
```

- **Pregunta 2: (10 puntos)** Rutee el siguiente programa.

```

#include <stdio.h>
void f(int* , int*);
int main(){
    int a = 5 , b = 8;
    int *p = NULL;
    a += b++;
    printf(" %d ", a);
    printf(" %d ", b);
    p = &a;
    f( p,&b);
    printf("%d %d", a, b);
    return 0;
}

void f(int *d,int *a){
    int z = 2, i = 300;
    for( i = 0 ; i < 20 ; i++){
        i += z;
        (*d) += i;
        return;
    }
    (*d) = (*a) + (*d);
}

```

Función main:

a	b	c
127 5	842 8	278 NULL
13	9	127
15		

Funcion f:

*d	*a	z	i
203 127	130 842	710 2	848 300
			0
			2

Pantalla:

13 9 15 9

■ Pregunta 3: (15 puntos)

```
#include <stdio.h> //Archivos de cabecera
int g(int* , int); //declaración de funcion g
int h(int , int*); //declaración de funcion h
int main(){ //Comienzo del main
    int *a = NULL, *c = NULL; //Declaración de los punteros 'a' y 'c'
    int b = a, d; //Declaración de variables
    //ERROR: Al asignar 'a' a b a pesar de que el compilador lo
    //deja pasar, no se debe realizar dado que
    //no tiene sentido
    d = 3; //Asignación de 3 a 'd'
    c = d; //Asignacion de 'd' a 'c' ERROR FALTÓ EL '&',
    //a pesar de que el compilador lo considera bueno
    //esta malo
    printf("%d\n", b); //Impresión de 'b' sale por pantalla un -> 0
    printf("%d\n", c); //Impresión de 'c' sale por pantalla un -> 3
    a = (&b); //se asigna la direccion de memoria de 'b' al puntero 'a'
    d = g(a,d); //se llama a la función 'g' con los valores de 'a' y 'd'
    //y el resultado es devuelto en d
    printf("%d\n", d); //Impresión de 'd' sale por pantalla un -> 5
    printf("%d\n", b); //Impresión de 'b' sale por pantalla un -> 4
    printf("%d\n", h( b, &d)); //Impresión de 'h' que es una función, la cual se le pasan
    //por parametro los valores de 'b' y la dirección de memoria
    //de 'd' y sale por pantalla un -> 0
    return 0; //termina la función main retornando un 0. declarando que
    // termino bien la función
}
int g(int *w,int b){ //función g recibe como parametros el puntero 'w' y el
    //entero 'b'
    int i; //declara i
    for(i = 1 ; i < 5 ; ++i) //Comienza el for en el valor '1', e incrementa antes de usar
    //el valor, y llega hasta que el numero 'i' sea menor a '5'
        (*w) += i; //Dado que no hay paréntesis en el for se realiza la
    //primera línea
    //en la cual el puntero 'w' se le suma las iteraciones de i,
    //pero dado
    //que se trabaja de la forma '(*w)' el cambio ocurre en
    //la variable
        (*w) = ++b; //se incrementa b y luego se suma a lo que apunta 'w'
    return i; //se retorna el valor de i
}
int h(int a,int *b){ //Funcion h recibe como parámetros un entero 'a' y un puntero
    //a entero 'b'
    return (a- ++(*b)); //Incrementa el valor de lo apuntado por 'b' y se lo
    //resta a 'a'
    //ese valor es el retornado
}
```