Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Departamento de Sistemas e Informática
Escuela de Electrónica
Informática Aplicada

Práctica 0: Repaso de C básico

Contenido:
Esta práctica está diseñada para que el estudiante recuerde los conceptos básicos del lenguaje C vistos en Informática.

1) Realizar un programa que declare las variables x, y, z, les asigne los valores 10, 20 y 30 e intercambie entre si sus valores de forma que el valor de x pasa a y, el de y pasa a z y el valor de z pasa a x (se pueden declarar variables auxiliares aunque se pide que se use el menor número posible).

2) Escriba un programa que lea por teclado diez números enteros distintos de cero y a continuación lea una secuencia de valores enteros indicando si están entre los diez valores leídos. Cuando se lea el valor cero, el programa finalizará.

Solución:

```
#include <stdio.h>

int main()
{
   int vleidos[10], valor,i;
   for (i=0; i<10; i++){
         scanf(" \n %d",&valor);
         vleidos[i]=valor;
         printf ("El valor leido es: %d \n",valor);

   }
   while (valor != 0){
         printf ("Ingrese un valor \n");
         scanf(" \n %d",&valor);
         for (int j=0; j<10; j++){
               if (valor==vleidos[j])
                     printf (" Valor ocupa el lugar : %d del vector \n", j );
               else continue;
         }
   }
       return 0;
}
```

3) Se ingresa por teclado la cantidad de agua caída, en milímetros día a día durante un mes. Se pide determinar el día de mayor lluvia, el de menor y el promedio

4) Escriba

a) código C para calcular la traza de una matriz cuadrada de double. Se denomina traza de una matriz cuadrada a la suma de los elementos de su diagonal principal.

b) código C para determinar la matriz transpuesta de otra (conteniendo números de tipo double). Una matriz transpuesta de otra, es aquella que tiene los mismos elementos pero dispuestos en forma distinta. Las columnas de la matriz original se transforman en filas de la matriz transpuesta.
c) código C para calcular el producto de dos matrices cuadradas de dimensión n.

5) Dada una frase en una cadena, mostrar en pantalla cada palabra que la compone:

a) sin usar las funciones estándar.
b) usando las funciones estándar.

6) Escriba un programa para ver las longitudes y valores máximos y mínimos en bytes de los tipos básicos de programación en C en su máquina:

Solución:

```
#include <stdio.h>
#include <limits.h>
#include <float.h>
int main(){
char a;
short int b;
int c;
long int d;
unsigned char e;
unsigned short int f;
unsigned int g;
unsigned long int h;
float i;
double j;
long double k;
printf ("Longitud de cada uno de los tipos basicos \n\n");
printf ("La longitud de char a= %d\n",sizeof(a));
printf ("La longitud de short int b= %d\n",sizeof(b));
printf ("La longitud de int c= %d\n",sizeof(c));
printf ("La longitud de long int d= %d\n",sizeof(d));
printf ("La longitud de unsigned char e= %d\n",sizeof(e));
printf ("La longitud de unsigned short int f= %d\n",sizeof(f));
printf ("La longitud de unsigned int g= %d\n",sizeof(g));
printf ("La longitud de unsigned long int h= %d\n",sizeof(h));
printf ("La longitud de float i= %d\n",sizeof(i));
printf ("La longitud de double j= %d\n",sizeof(j));
printf ("La longitud de long double k= %d\n",sizeof(k));
printf("\nValores minimos y maximos de cada uno de los tipos\n\n");
printf ("Minimo y maximo de char a= %d\t\t%d\n",CHAR_MIN,CHAR_MAX);
printf ("Minimo y maximo de short int b=%d\t\t%d\n",SHRT_MIN,SHRT_MAX);
printf ("Minimo y maximo de int c= %d\t\t%d\n",INT_MIN,INT_MAX);
printf ("Minimo y maximo de long int d=%d\t\t%d\n",LONG_MIN,LONG_MAX);
printf ("Maximo de unsigned char e= %d\n",UCHAR_MAX);
printf ("Maximo de unsigned short int f= %d\n",USHRT_MAX);
printf ("Maximo de unsigned int g= %d\n",UINT_MAX);
printf ("Maximo de unsigned long int h= %d\n",ULONG_MAX);
printf ("Minimo y maximo de float i= %d\t\t%d\n",FLT_MIN,FLT_MAX);
printf ("Minimo y maximo de double j= %d\t\t%d\n",DBL_MIN,DBL_MAX);
printf ("Minimo y maximo de long double k=%d\t\t%d\n",LDBL_MIN,LDBL_MAX);
}
```

7) Escribir una directiva de preprocesador para realizar cada una de las siguientes tareas:
a. Si la constante simbólica TRUE está definida, eliminarla y volverla a definir como 1.

b. Ídem pero sin usar la directiva #ifdef .
c.Ídem pero usando otra directiva de compilación condicional distinta de las 2 anteriores.

Ayuda para directivas del preprocesador:
http://www.itlalaguna.edu.mx/academico/carreras/sistemas/programacion2/cpp4.pdf
http://mondrian.die.udec.cl/~mmedina/Clases/LenProg/2008-1/ALP18-Preprocesador.pdf
http://web.fi.uba.ar/~bortega/apunte_c_a_bajo_nivel.pdf
Kernighan/Ritchie 4.11 El preprocesador de C

8) Escribir una macro de función "MINIMO2" que determine el menor de dos números pasados como argumento. Escribir un programa que use una macro MINIMO3 para determinar el menor de tres valores numéricos. Dicha macro deberá utilizar la macro MINIMO2.

9) Dadas las siguientes macros:

```
//pone a 1 el bit b del número a
#define BitSet(a,b)       ( (a) |=  ( 1<<(b) ) )
//pone a 0 el bit b del número a
#define BitClear(a,b)     ( (a) &= ~( 1<<(b) ) )
//verifica si el bit b del número a es un 0 o un 1
#define BitCheck(a,b)     ( (a) &   ( 1<<(b) ) )
//cambia el bit b del número a por su valor complementario
//si el bit es 0 lo pone a 1 y si es 1 lo pone a 0
#define BitToggle(a,b)    ( (a) ^=  ( 1<<(b) ) )
```

En una función `main` dar ejemplos de utilización de cada una de ellas utilizando 2 números de tipo `unsigned int`.
Utilizar la siguiente función para visualizar en pantalla un entero `number` en formato binario:

```
void print_binary(int number)
{
    if (number) {
        print_binary(number >> 1);
        putc((number & 1) ? '1' : '0', stdout);
    }
}
```

Comparar con lo que imprime:

```
char buffer[33];
printf("%s\n",itoa(number,buffer,2));//incluir stdlib.h
```

10) Escribir la función paridad_par que recibe un byte y retorna un byte. El byte devuelto debe ser igual al byte recibido o, igual al byte recibido con el bit más significativo modificado, de forma tal que la configuración del byte entregado contenga un número par de unos.

11) Generar una función `unsigned rightRot(unsigned x,int n)` que rote a derecha los últimos n bits de x. Asumir que el bit menos significativo de x, ocupa la posición cero y que n siempre asume valores positivos.

Solución:

```
unsigned rightRot(unsigned x,int n){
   int i;
   for(i =0; i<n; i++){
         if ((x & 1) == 0 ) //el ultimo bit es cero
               x = x >> 1;
         else {             //el ultimo bit es 1
               x = x >> 1;
               x = x | 0x80;
         }
```

```
    }
    return x;
}


int main()
{
    unsigned int x = 0xB7;
    unsigned int xx= rightRot(x,5);
    return 0;
}
```

12) Generar una función `unsigned setbits(unsigned x, int p, int n, unsigned y)` que regresa `x` con los `n` bits que principian en la posición `p` iguales a los `n` bits más a la derecha de `y`, dejando los otros bits sin cambio.
Nota: Ejercicio del Kernighan y Ritchie resuelto y explicado en: http://hitmontop-ejerciciosresueltos.blogspot.com.ar/

13) Generar una función `unsigned getBits(unsigned x,int p, int n)` que retorne los `n` bits de `x` (ajustado a la derecha) que principia en la posición `p`. Se supone que la posición del bit 0 está en el borde derecho y que `n` y `p` son valores positivos adecuados. Por ejemplo, `getbits(x,4,3)` regresa los 3 bits que están en la posición 4, 3 y 2, ajustados a la derecha.
(ver si puede sintetizar la operación en una sola instrucción).
Nota: Ejemplo del Kernighan y Ritchie (Sección 2.9) cuya resolución está explicada en: http://hitmontop-ejerciciosresueltos.blogspot.com.ar/

14) Escriba un programa que intercambie el valor de dos enteros `num1` y `num2` sin utilizar variables temporales, usando el operador XOR.

15) Escriba una función que determine si existe, por lo menos un bit entre las posiciones baja y alta (con baja<=alta) con valor 1, de un número entero llamado num pasado a dicha función como argumento. Si es así dicha función debe retornar un 1, en caso contrario retornará 0.

16) Para un programa que hará uso del puerto paralelo de una PC se necesita disponer de funciones que permitan colocar a nivel lógico 1 o 0, un bit determinado dentro de un byte a ser presentado al puerto paralelo. Se pide:
a) Escribir una función llamada `Set()`, que obedezca al siguiente prototipo:

```
unsigned char Set(unsigned char Datos, short Linea);
```

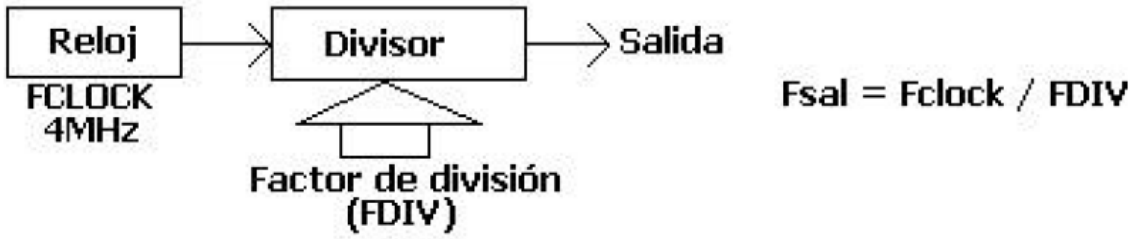
que reciba un byte sobre la variable `datos` y el número de línea que será forzada a nivel lógico 1, devolviendo por el nombre el resultado de la operación.
Ejemplo: Si en `datos` se recibe 1000 0001 y `Linea = 3`, se debe retornar el byte 1000 1001.
b) Dar un ejemplo de invocación de la función.
c) Ídem a) pero para una función llamada `Clear()` que coloque a nivel lógico 0 la línea indicada, y devuelva por el nombre el resultado de la operación.
Ejemplo: Si en `Datos` se recibe 0111 1100 y `Linea = 3`, se debe devolver el byte 0111 0111.
d) Dar un ejemplo de invocación de la función.
ACLARACION: La línea 0 se corresponde con el bit menos significativo (LSB) del byte, y la linea 7 con el bit más significativo (MSB) del byte.

16) Un sintetizador de una radio digital posee un generador de frecuencia controlado por el siguiente registro:

bit 7	6	5	4	3	2	1	bit 0
AFT	BAND	E	D	C	B	A	SYN

AFT: Control de Sintonía Fina Automática (1 = ON , 0 = OFF).
BAND: Control de selección de banda (1 = AM , 0 = FM).
E, D, C, B, A: Factor de división de la frecuencia de reloj (ver tabla). FE DE ERRATAS: EN LA COLUMNA FDIV DONDE DICE 30 DEBERÍA DECIR 31 Y DONDE DICE 31 SERÍA 32).
SYN: Control de activación del sintetizador (1 = ON, 0 = OFF).
SYN: Control de activación del sintetizador (1 = ON, 0 = OFF).

E	D	C	B	A	Factor de división (FDIV)
0	0	0	0	0	1
0	0	0	0	1	2
0	0	0	1	0	3
0	0	0	1	1	4
...	...	...	...	...	...
1	1	1	1	0	30
1	1	1	1	1	31



Se pide:
a) Defina un tipo enumerativo denominado "band" con los símbolos AM y FM, para representar las bandas del sintonizador.
b) Escriba el código de una función denominada "`get_synthesizer_divider()`" que devuelva por su nombre el factor de división a partir de los bits A, B, C, D y E contenidos en el byte de control. El prototipo de la función pedida es:

```
int get_synthesizer_divider (unsigned char);
```

c) Dar un ejemplo de invocación de la función.
d) Escriba el código de una función denominada "`get_band()`" que reciba como parámetro el registro de control, y retorne por el nombre la banda seleccionada como un tipo enumerativo "band". Dar un ejemplo de invocación.
NOTA: Debe utilizarse máscaras y operadores de bits para operar con los bits del registro de datos.

17) Para un aplicativo a ejecutarse sobre una plataforma de 32 bits que contendrá una interfaz gráfica de usuario (GUI), se requiere disponer de funciones para el tratamiento de colores de los elementos gráficos de la interfaz.
El color de cualquier elemento gráfico se almacenará en una variable de tipo `int` sin signo, de acuerdo a la convención RGB (ejemplo: un color con representación RGB = 0xFFCCAA se

corresponde con 0xFF, 0xCC y 0xAA para las componentes Rojo, Verde y Azul, respectivamente). Se pide:
a) Escribir una función denominada "`Rojo()`" que reciba como argumento el color en representación RGB sobre una variable de tipo `int` sin signo, y que devuelva por el nombre la componente de color rojo como un `char` sin signo. Dar un ejemplo de invocación de la función.
b) Idem a) para una función denominada "`Verde()`" que devuelva la componente verde. Dar un ejemplo de invocación de la función.
c) Idem a) para una función denominada "`Azul()`" que devuelva la componente Azul. Dar un ejemplo de invocación de la función.
d) Escribir una función llamada `componentesRGB()` que reciba como argumentos el color en notación RGB e indique cada componente presentes en dicho color.

Referencias:

1. Algunos ejercicios se extrajeron de: http://materias.fi.uba.ar/7502E/material.html