

Informática II Practica No. 2

1. Objetivos

- Familiarizarse con la notación binaria y hexadecimal para representar números enteros.
- Introducir el concepto de dirección de memoria, arreglos y apuntadores en el lenguaje C/C++.
- Definición e implementación de funciones.
- Crear y usar librerías.
- Introducir Arduino y Tinkercad.

2. Ejercicios

Ejercicio 1. Codifique el programa que se muestra a continuación:

```
#include <iostream>
using namespace std;

void fun_a(int *px, int *py);
void fun_b(int a[], int tam);

int main()
{
    int array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    fun_b(array, 10);
}

void fun_a(int *px, int *py) {
    int tmp = *px;
    *px = *py;
    *py = tmp;
}

void fun_b(int a[], int tam) {
    int f, l;
    int *b = a;
    for (f = 0, l = tam - 1; f < l; f++, l--) {
        fun_a(&b[f], &b[l]);
    }
}
```

Con ayuda del debugger, examine la representación en memoria del arreglo array y responda las siguientes preguntas.

- Cuál es su dirección en memoria? ¿Cuántos bytes se dedican para almacenar cada elemento de array?
- Cuál es la dirección y el contenido en memoria del elemento array[3] ?
- Describa el efecto que tiene la función fun_b, sobre el arreglo array.

Ejercicio 2. La función que se ilustra a continuación calcula el promedio de los elementos de un arreglo de tamaño n. promedio y suma apuntan a variables que la función modifica por referencia. Desafortunadamente, la función contiene errores; encuétralos y corrígelos, de tal manera que su operación sea correcta.

```
void fun_c(double *a, int n, double *promedio, double *suma){
    int i;
    suma = 0.0;
    for (i = 0; i < n; i++)
        suma += (a + i);

    promedio = suma / n;
}
```

Implemente un caso de prueba para la función anterior, que le permita verificar su correcto funcionamiento.

Ejercicio 3. Se tiene la siguiente declaración e inicialización para el arreglo b:

```
unsigned short b[4][2] = {{77, 50}, {5, 2}, {28, 39}, {99, 3}};
```

La Figura 1, que se presenta a continuación, ilustra la representación del arreglo b en memoria. Complete la numeración de las direcciones de memoria, en notación hexadecimal, para cada elemento del arreglo b.

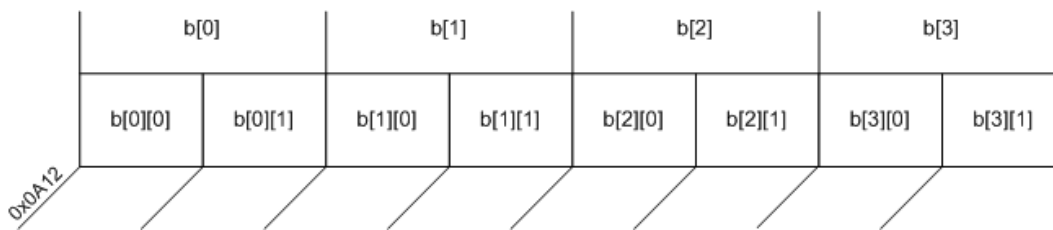


Figura 1: Elementos del arreglo b.

Determine de acuerdo a lo ilustrado en la Figura 1 el valor correspondiente de las siguientes expresiones:

- b
- b+2
- *(b+2)
- *(b+2)+1
- (*(b+2)+1)
- b[3][1]
- *b++

Ejercicios con Arduino

Ejercicio 4. Realice un circuito en Tinkercad donde la intensidad de iluminación de un LED se pueda variar con un potenciómetro. Debe agregar en Tinkercad una *Breadboard Small*, un *Potenciometer*, un *Resistor* de 220Ω , un *LED*, y un Arduino. Conecte todos los elementos del circuito de una manera apropiada y programe el Arduino para la tarea pedida. Considere usar el montaje de la figura 2.

Ejercicio 5. Realice un circuito en Tinkercad para escribir mensajes en un *Liquid-Crystal Display* (LCD) con la ayuda de Arduino. Cambie de *Basic* a *All* en la opción *components* de Tinkercad. Debe agregar un *LCD 16 × 2*, y un Arduino. Conecte todos los elementos del circuito de una manera apropiada y programe el Arduino para la tarea pedida. Considere el esquema de la Figura 3 y el montaje de la Figura 4.

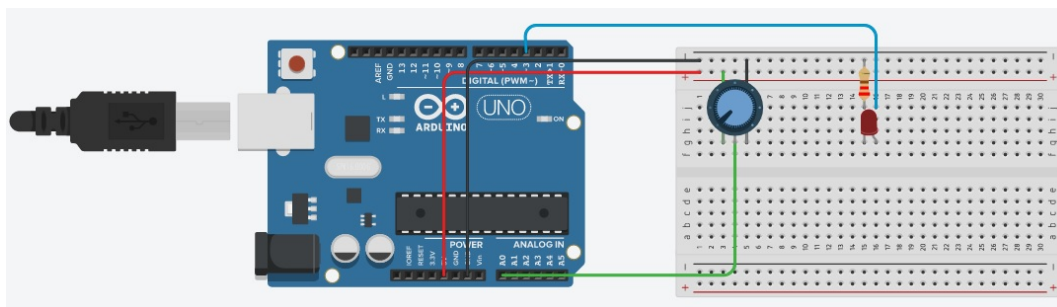


Figura 2: Montaje en protoboard del potenciómetro.

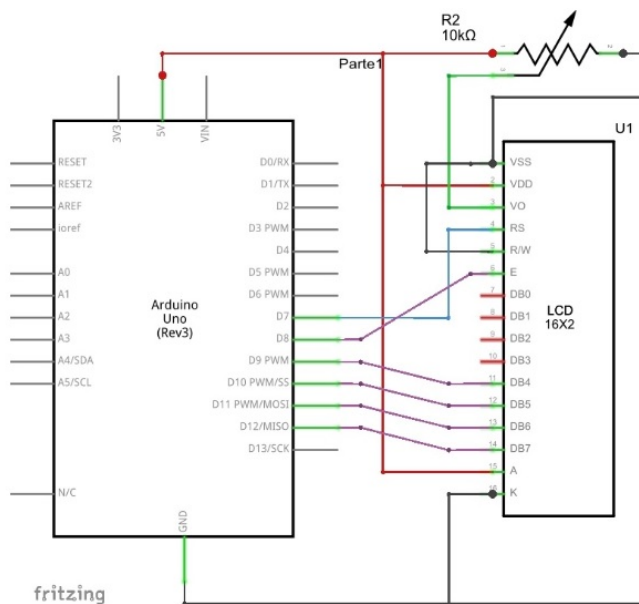


Figura 3: Esquemático Liquid-Crystal Display.

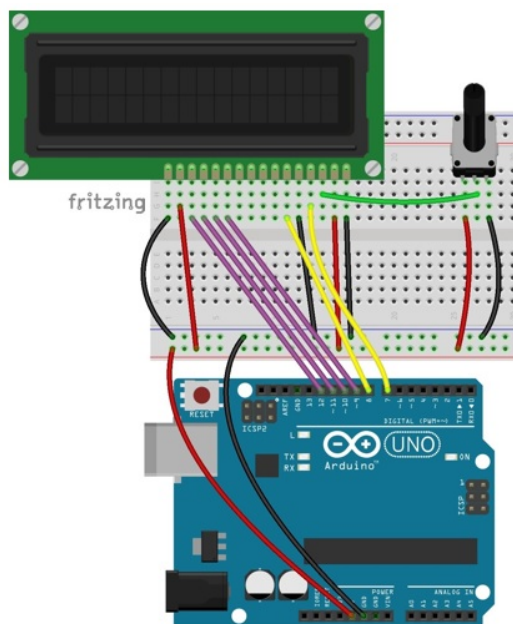


Figura 4: Montaje Liquid-Crystal Display.

3. Problemas

Problema 1. Se necesita un programa que permita determinar la mínima combinación de billetes y monedas para una cantidad de dinero determinada. Los billetes en circulación son de \$50.000, \$20.000, \$10.000, \$5.000, \$2.000 y \$1.000, y las monedas son de \$500, \$200, \$100 y \$50. Hacer un programa que entregue el número de billetes y monedas de cada denominación para completar la cantidad deseada. Si por medio de los billetes y monedas disponibles no se puede lograr la cantidad deseada, el sistema deberá decir lo que resta para lograrla. Use arreglos y ciclos para realizar el programa.

Ejemplo: si se ingresa 47810, el programa debe imprimir:

```
50000 : 0
20000 : 2
10000 : 0
5000 : 1
2000 : 1
1000 : 0
500 : 1
200 : 1
100 : 1
50 : 0
Faltante: 10
```

Realice una versión en Arduino de este programa (en un Arduino físico o Tinkercad), la cantidad de dinero debe ser ingresada con la ayuda del serial. Use el monitor serial de Arduino o Tinkercad para ingresar los valores necesarios e imprimir.

Problema 2. Elabore un programa que genere un arreglo de 200 letras mayúsculas aleatorias, imprima este arreglo y luego imprima cuantas veces se repite cada letra en el arreglo.

Ejemplo: supóngase que se genera el arreglo de 10 elementos: ABARSECAAB. El programa debe imprimir: ABAR-SECAAB

```
A: 4
B: 2
C: 1
Y así sucesivamente.
```

Problema 3. Haga una función que compare 2 cadenas de caracteres y retorne un valor lógico verdadero si son iguales y falso en caso contrario, no olvide también tener en cuenta la longitud de las cadenas. Escriba un programa de prueba.

Problema 4. Haga una función que reciba una cadena de caracteres numéricos, la convierta a un número entero y retorne dicho número. Escriba un programa de prueba.

Ejemplo: si recibe la cadena “123”, debe retornar un int con valor 123.

Problema 5. Haga una función que reciba un numero entero (int) y lo convierta a cadena de caracteres. Use parámetros por referencia para retornar la cadena. Escriba un programa de prueba.

Ejemplo: si recibe un int con valor 123, la cadena que se retorne debe ser “123”.

Problema 6. Escriba un programa que reciba una cadena de caracteres y cambie las letras minúsculas por mayúsculas, los demás caracteres no deben ser alterados.

Ejemplo: se recibe Man-zana debe mostrar MAN-ZANA.

Nota: la salida del programa debe ser: Original: Man-zana. En mayuscula: MAN-ZANA.

Realice una versión en Arduino de este programa (en un Arduino físico o Tinkercad). Los datos deben ser ingresados con la ayuda del serial. Use el monitor serial de Arduino o Tinkercad para ingresar los valores necesarios e imprimir.

Problema 7. Escriba un programa que reciba una cadena de caracteres y elimine los caracteres repetidos.

Ejemplo: se recibe bananas debe mostrar bans.

Nota: la salida del programa debe ser: Original: bananas. Sin repetidos: bans.

Problema 8. Escriba un programa que reciba una cadena de caracteres y separe los números del resto de caracteres, generando una cadena que no tiene números y otra con los números que había en la cadena original.

Ejemplo: Si se recibe abc54rst el programa debe imprimir las cadenas: abcrst y 54.

Nota: la salida del programa debe ser:

Original: abc54rst.

Texto: abcrst. Numero: 54.

Problema 9. Escribir un programa que reciba un número n y lea una cadena de caracteres numéricos, el programa debe separar la cadena de caracteres en números de n cifras, sumarlos e imprimir el resultado. En caso de no poderse dividir exactamente en números de n cifras se colocan ceros a la izquierda del primer número.

Ejemplo: Si n=3 y se lee el arreglo 87512395 la suma sería $087+512+395=994$.

Nota: la salida del programa debe ser:

Original: 87512395.

Suma: 994.

Problema 10. Escribir un programa que permita convertir un número en el sistema romano al sistema arábigo usado actualmente. A continuación se encuentran los caracteres usados en el sistema romano y su equivalente arábigo:

M: 1000

D: 500

C: 100

L: 50

X: 10

V: 5

I: 1

Los números romanos se forman usando estos caracteres en base a dos reglas:

- Si un carácter está seguido por uno de igual o menor valor, su valor se suma al total.
- Si un carácter está seguido por uno de mayor valor, su valor se resta del total.

Ejemplo: CC=200, CD=400, DC=600, DCLXVI=666, CLXXIV=174.

Nota: la salida del programa debe ser:

El número ingresado fue: DCLXVI

Que corresponde a: 666.

Realice una versión en Arduino de este programa (en un Arduino físico o Tinkercad). Los datos deben ser ingresados con la ayuda del serial. Use el monitor serial de Arduino o Tinkercad para ingresar los valores necesarios e imprimir.

Problema 11. Escriba un programa que permita manejar las reservas de asientos en una sala de cine, los asientos de la sala de cine están organizados en 15 filas con 20 asientos cada una. El programa debe mostrar una representación de la sala que indique que asientos están disponibles y cuales se encuentran reservados. Además debe permitir realizar reservas o cancelaciones al ingresar la fila (letras A-O) y el número del asiento (números 1-20).

Nota: un ejemplo de visualización de una sección de la sala es el siguiente:

+	+	+	+
-	-	+	+
-	-	-	-

Donde + representa los asientos reservados y - representa los asientos disponibles.

Problema 12. Un cuadrado mágico es una matriz de números enteros sin repetir, en la que la suma de los números en cada columna, cada fila y cada diagonal principal tienen como resultado la misma constante. Escriba un programa que permita al usuario ingresar una matriz cuadrada, imprima la matriz y verifique si la matriz es un cuadrado mágico.

Nota: un ejemplo de cuadrado mágico es el siguiente:

4	9	2
3	5	7
8	1	6

Problema 13. Se tiene una fotografía digitalizada de una porción de la galaxia NGC 1300 que está ubicada a 61.000.000 de años luz del planeta Tierra. La representación digital de la imagen está constituida por una matriz de números enteros; en la cual, cada uno representa la cantidad de luz en ese punto de la imagen, así:

0	3	4	0	0	0	6	8
5	13	6	0	0	0	2	3
2	6	2	7	3	0	10	0
0	0	4	15	4	1	6	0
0	0	7	12	6	9	10	4
5	0	6	10	6	4	8	0

Se puede determinar si el elemento $a_{i,j}$ de la matriz representa una estrella si se cumple que:

$$\frac{a_{i,j} + a_{i,j-1} + a_{i,j+1} + a_{i-1,j} + a_{i+1,j}}{5} > 6 \quad (1)$$

Elabore y pruebe una función que reciba un puntero a la matriz de enteros como argumento y que retorne el número de estrellas encontradas en la imagen. Ignore las posibles estrellas que puedan existir en los bordes de la matriz.

Problema 14. Elabore un programa que llene una matriz 5x5 con los números del 1 al 25 y la imprima, luego imprima la matriz rotada 90, 180 y 270 grados.

Ejemplo:

Matriz Original					Matriz Rotada 90°				
1	2	3	4	5	21	16	11	6	1
6	7	8	9	10	22	17	12	7	2
11	12	13	14	15	23	18	13	8	3
16	17	18	19	20	24	19	14	9	4
21	22	23	24	25	25	20	15	10	5

Problema 15. Elabore un programa que permita hallar la intersección entre un par de rectángulos. Represente los rectángulos como arreglos de 4 datos de la siguiente manera:

- Los primeros 2 datos corresponden a las coordenadas de la esquina superior izquierda del rectángulo (x,y) como se observa en la Figura 5.
- Los siguientes 2 datos representan el ancho y la altura del rectángulo como se observa en la Figura 5.

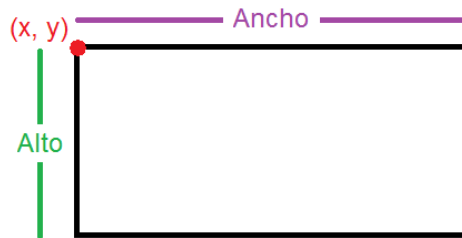


Figura 5: Parámetros del rectángulo.

Implemente una función que reciba 2 arreglos que representen los rectángulos A y B, y por referencia retorne un rectángulo C (con la misma estructura descrita anteriormente) que corresponda a la intersección de A y B como se observa en la Figura 6.

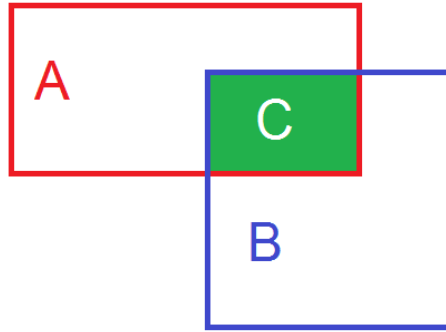


Figura 6: Elementos del arreglo b.

Ejemplo: Si se ingresan los rectángulos A y B representados por los arreglos $\{0, 0, 8, 4\}$ y $\{5, 2, 6, 7\}$, el rectángulo intersección C debe ser el arreglo $\{5, 2, 3, 2\}$.

Problema 16. En una malla de 2×2 , realizando únicamente movimientos hacia la derecha y hacia abajo hay 6 posibles caminos para llegar de la esquina superior izquierda a la inferior derecha como se observa en la Figura 7.

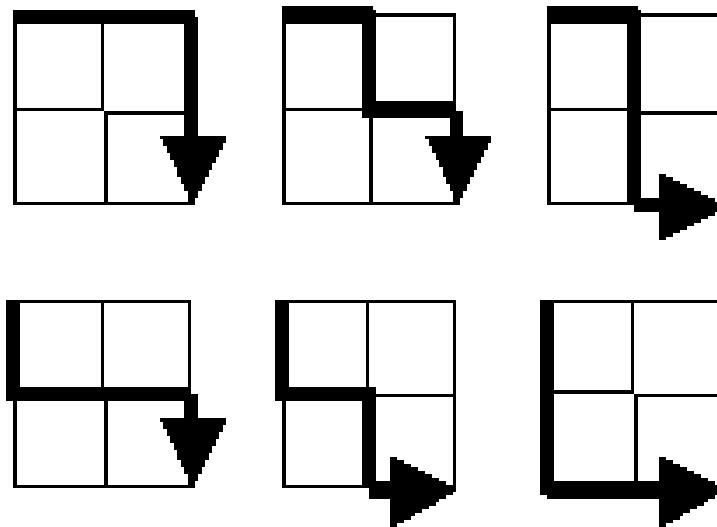


Figura 7: Caminos posibles en una malla 2×2 .

Escriba un programa que reciba un número n y calcule el número de caminos posibles en una cuadrícula de $n \times n$.

Nota: la salida del programa debe ser:

Para una malla de 2×2 puntos hay 6 caminos.

Problema 17. Dos números a y b ($a \neq b$) son amigables si la suma de los divisores de a (excluyéndose el mismo) es igual a b , y viceversa. Ej: los divisores de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110; y suman 284. Los divisores de 284 son 1, 2, 4, 71 y 142; y suman 220. Entonces 220 y 284 son amigables. Escribir un programa que reciba un número y halle la suma de todos los números amigables menores al número ingresado.

Nota: la salida del programa debe ser:

El resultado de la suma es: 504.

Realice una versión en Arduino de este programa (en un Arduino físico o Tinkercad). Los datos deben ser ingresados con la ayuda del serial. Use el monitor serial de Arduino o Tinkercad para ingresar los valores necesarios e imprimir.

Problema 18. Las permutaciones lexicográficas son permutaciones ordenadas numérica o alfabéticamente, por

ejemplo las permutaciones lexicográficas de 0,1 y 2 son: 012, 021, 102, 120, 201, 210. Escribir un programa que reciba un número n y halle la n -ésima permutación lexicográfica de los números entre 0 y 9.

Ejemplo: para $n=1000000$, la permutación lexicográfica es 2783915460.

Nota: la salida del programa debe ser:

La permutacion numero 1000000 es: 2783915460.

Problema de bonificación. Realice un programa con un Arduino físico para simular ciertos botones del teclado del computador. Cuando se presione un pulsador físico se debe escribir en pantalla alguna letra. Ponga al menos 4 pulsadores para las letras 'W', 'A', 'S', y 'D'.

Nota: Considere el uso de la librería *Keyboard.h* de Arduino.