

GUIA DE EJERCICIOS

Wenceslao Palma <wenceslao.palma@ucv.cl>

1 Iteración, selección

1. Escriba un programa en lenguaje C que genere las primeras N filas de:

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
...
```

2. Un número entero se puede expresar como un producto de números primos. Dicha expresión se llama descomposición de un número en factores primos. Escriba un programa en lenguaje C que calcule la descomposición de un número en factores primos. Por ejemplo: $13860 = 2 \times 2 \times 3 \times 3 \times 5 \times 7 \times 11$
3. El algoritmo de Newton, usado para calcular la raíz cuadrada, se basa en aproximaciones sucesivas donde la primera aproximación de la raíz de un número N es:
$$T_1 = N/2$$
$$T_{i+1} = T_i/2 + N/(2 * T_i)$$

Escriba un programa en lenguaje C, que permita determinar la raíz cuadrada de un número en base a la recurrencia anterior.

4. Dado un número entero positivo, su crápulo es un número que se obtiene de la sgte forma: se suman los dígitos que lo componen y si el valor de la suma es menor que 10, el crápulo es el valor obtenido sino el crápulo es el crápulo de la suma de los dígitos. Escriba un programa en lenguaje C que lea un entero positivo y escriba el valor de su crápulo.
5. Un número entero n se dice perfecto si la suma de sus divisores es igual a n. Si la suma de los divisores es mayor que n, el número se dice abundante. Si dicha suma es menor que n, el número se dice deficiente. Escriba un algoritmo que determine si un número es abundante, deficiente o perfecto.
6. Cuando la suma de los dígitos alternos de un número son iguales, ese número es divisible por 11. Por ejemplo: 5841 . $5+4=8+1$. Escriba un programa en lenguaje C que determine si un número entero positivo es divisible por 11.
7. Un número se dice automórfico si su cuadrado termina en los mismos dígitos que el número original, por ejemplo $76^2 = 5776$. Un número se

dice trimórfico si su cubo termina en los mismos dígitos que el número original, por ejemplo $49^3 = 117649$. Escriba un programa en lenguaje C que determine los números automórficos y trimórficos menores que 1000.

8. La multiplicación rusa consiste en multiplicar sucesivamente por 2 el multiplicando y dividir por 2 el multiplicador hasta que el multiplicador tome el valor 1. Luego, se suman todos los multiplicandos correspondientes a los multiplicadores impares. Dicha suma es el resultado del producto de los dos números. Por ejemplo, para multiplicar 37 por 12 el resultado final es $12 + 48 + 384 = 444$.
9. Escriba un programa en lenguaje C que permita obtener la suma de todos los números que se puedan generar al permutar n dígitos consecutivos a partir del 1. El número de permutaciones se define como todos los números distintos que se pueden formar cambiando la posición de los dígitos. La cantidad de permutaciones de números con n dígitos es $n! = 1 \times 2 \times 3 \times 4 \times \dots \times n$. El siguiente algoritmo permite realizar el proceso en forma rápida:

- Cada dígito estará $n!/n$ veces en el total de números obtenidos de las permutaciones.
- A continuación se suman los n dígitos y se multiplica este resultado por el número de veces obtenido en el paso anterior.
- El número obtenido en el paso 2 se suma n veces, siendo n el número de dígitos. Cada sumando debe desplazarse una posición.

Ejemplo: Considerando $n=3$, el número será 123. El número de veces que se repite cada dígito es $3!/3 = 2$. La suma de los dígitos es $1 + 2 + 3 = 6$. La multiplicación de la suma de los dígitos por el número de veces que se repite cada dígito es $2 \times 6 = 12$. Como son 3 dígitos debe sumarse 3 veces 12 con un desplazamiento cada vez, es decir:

$$\begin{array}{r}
 12 \\
 12 \\
 + 12 \\
 \hline
 1332
 \end{array}$$

10. Un número entero positivo se dice M-alternante si: El dígito menos significativo es par (impar) entonces los 2 dígitos siguientes deben ser impares (pares), luego los tres siguientes dígitos deben ser pares (impares) y así sucesivamente. Por ejemplo : 357221 es un número M-alternante de orden 3. Escriba un programa que determine si un número es o no M-alternante y si lo es a que orden corresponde.
11. Un anillo primo esta compuesto de n dígitos (n par), ubicados uno al lado del otro, y es tal que la suma de los dígitos adyacentes siempre es un número primo. Por ejemplo : 143256 corresponde a un anillo primo ya que:

$1+4 = 5$
 $4+3 = 7$
 $3+2 = 5$
 $2+5 = 7$
 $5+6 = 11$
 $6+1 = 7$

Escriba un programa en C que verifique si un número ingresado cumple con la condición de estar compuesto por una serie de dígitos que lo califican como un anillo primo.

- Un par de números m y n son llamados par amigable, si la suma de todos los divisores de m (excluyendo m) es igual al número n y la suma de todos los divisores del número n (excluyendo n) es igual a m ($m! = n$). Por ejemplo, los números 220 y 284 son un par amigable porque los únicos números que dividen de forma exacta a 220 son 1,2,4,5,10,11,20,22,44,55 y 110, y $1 + 2 + 4 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$. Los números que dividen a 284 son 1,2,4,71 y 142, y $1 + 2 + 4 + 71 + 142 = 220$. Por lo tanto 220 y 284 son un par amigable. Escriba un programa en C que ingrese m y n , asegure que m es distinto de n y determine si dichos números son par amigable.

2 Arreglos unidimensionales/bidimensionales

- Desarrolle un programa en lenguaje C que permita rotar una matriz en ángulos de 0,90,180 ó 270 grados. El programa debe: ingresar valores en una matriz de $N \times N$, el ángulo en que se desea rotar y mostrar la matriz rotada en el ángulo que corresponda. Por ejemplo:

```

1 2 rotación en 90° 2 4
3 4                  1 3

```

- Un número se dice circular si es posible recorrerlo de la siguiente manera: Se toma el dígito de más a la izquierda y su valor indicará la cantidad de veces que se debe contar hacia la derecha, esta "cuenta" se detiene en un dígito del número el cual también indicará la cantidad de veces que se debe contar hacia la derecha y así sucesivamente. Este proceso da origen a un número circular si todos los dígitos del número se consideran tan solo una vez (como número originador de un conteo) y además se retorna al dígito de más a la izquierda que fue el que comenzó el conteo.

Valide que no existan dígitos repetidos.

Ejemplo : Si el número es 81362

```

(1) 8 1 3 6 2
(2) 8 1 3 6 2
(3) 8 1 3 6 2
(4) 8 1 3 6 2
(5) 8 1 3 6 2
(6) 8 1 3 6 2

```

Este es un número circular!. Escriba un programa en lenguaje C que permita decidir si un número es circular.

3. Considere una secuencia arbitraria de números enteros. Es posible ubicar los operadores + y - entre los números de la secuencia, de este modo se obtienen diferentes expresiones aritméticas que producen diferentes valores. Por ejemplo para la secuencia: 17, 5, -21, 15 existen 8 posibles expresiones:

```
17 + 5 + -21 + 15 = 16
17 + 5 + -21 - 15 = -14
17 + 5 - -21 + 15 = 58
17 + 5 - -21 - 15 = 28
17 - 5 + -21 + 15 = 6
17 - 5 + -21 - 15 = -24
17 - 5 - -21 + 15 = 48
17 - 5 - -21 - 15 = 18
```

una expresión se dice divisible por K si los operadores + y - son ubicados de tal manera que dan origen a un valor divisible por K. Escriba un programa que determine la divisibilidad de una expresión. Los datos de entrada deben ser K ($2 \leq K \leq 100$) y la secuencia de N enteros ($1 \leq N \leq 1000$). La salida debe ser: La expresión xxxxxx es Divisible por K o ninguna expresión generada es divisible por K.

Por ejemplo:

Entrada: K=3 y la secuencia 17 5 21 15

Salida:

```
La expresión 17 - 5 + -21 + 15 = 6 es divisible por 3
La expresión 17 - 5 + -21 - 15 = -24 es divisible por 3
La expresión 17 - 5 - -21 + 15 = 48 es divisible por 3
La expresión 17 - 5 - -21 - 15 = 18 es divisible por 3
```

4. Un método clásico para encontrar todos los números primos, que se encuentran en una secuencia de números desde 2 hasta N, es la llamada Criba de Eratóstones. Para ello el método va marcando (eliminando) todos los múltiplos de 2, 3, 4, 5 y así sucesivamente hasta que se encuentre el primer número no marcado que es mayor que \sqrt{N} .
Ejemplo : suponga una secuencia desde 2 a 20.

- Se toma el número 2, ya que es el primer número no marcado.
- Marcar todos los múltiplos de 2 a partir de 2^2 .
- Se toma el 3 y se marcan todos los múltiplos de 3 a partir de 3^2 .
- Se toma el 5 pero como es mayor que 20 se detiene el proceso y todos los números de la secuencia que no fueron marcados corresponden a los números primos que existen entre 2 y 20.

Escriba un programa en C que mediante el uso de arreglos y el método descrito anteriormente determine todos los números primos entre 2 y N.

5. Considere un juego que contempla un jugador y un tablero de $M \times N$ casilleros. En cada casillero es ubicado un entero positivo en el rango de 0 hasta 9. El objetivo del juego es remover todos los números desde el tablero. El jugador realiza sus intentos seleccionando un casillero, de este modo todos los casilleros ubicados en la **región conectada** que contienen el mismo número que el casillero seleccionado son eliminados y todos los casilleros ubicados sobre los eliminados son reubicados hacia abajo del tablero. Si todos los casilleros de una columna han sido removidos entonces las columnas ubicadas hacia su derecha se desplazan hacia la izquierda. El juego termina cuando todas los casilleros son eliminados o cuando no es posible continuar el juego.

Una **región conectada** esta compuesta de todas los casilleros adyacentes que pueden ser alcanzados mediante un movimiento horizontal (izquierda o derecha) y/o vertical (arriba o abajo) y considerando que todos los casilleros en la **región conectada** contienen el mismo número. Por ejemplo:

```

1 3 5 2 2
2 2 3 5 1
1 2 3 5 5

```

Puede dar origen a las siguientes jugadas:

```

1 3 5
2 2 3 5 1
1 2 3 5 5

```

```

      5
1   3 5 1
1 3 3 5 5

```

```

1   5 1
1 5 5 5

```

```

1
1 1

```

Se debe indicar como dato de entrada la dimension del tablero, indicando el número de filas y columnas. El contenido del tablero consiste de enteros no negativos $\in [0..9]$ que serán almacenados por filas, desde la posición (0,0). Considere que un tablero puede tener un tamaño máximo de 10×10 . Además el usuario, debe ingresar cada jugada en términos de coordenadas relativas al tablero. Las coordenadas deben ser ingresadas usando el siguiente formato: fila columna

Cada vez que el usuario juegue (ingresando una coordenada) se deberá mostrar el tablero resultante.

6. En un arreglo bidimensional llamado Puzzle se encuentra una instancia de un puzzle 4×4 . Para determinar que tan lejos nos encontramos de la

solución se puede utilizar la sgte expresión:

$$D = \sum_{i=0}^{15} |filaOriginal_i - filaActual_i| + |columnaOriginal_i - columnaActual_i|$$

Escriba un programa en C que permita calcular D . No se permite el uso de arreglos auxiliares.

3 Funciones

1. Un número es un palíndrome si la secuencia de dígitos que lo componen es idéntica al leerla desde izquierda a derecha y viceversa. Ahora, al sumar un número como por ejemplo 65 con el número que da como resultado el leerlo desde la derecha hacia la izquierda se genera 121, el cual es palíndrome. Si se realiza lo mismo con otro número, llegará un momento en el cual se generará un palíndrome.

Escriba la función **int cantidadSumas(long int)** que dado un número entero, entregue como resultado la cantidad de sumas necesarias para generar un palíndrome.

Por ejemplo:

```
Si n=47
47+74 = 121 --> 1
Si n =143
143+341 = 484 --> 1
Si n=87
87+78 = 165
165+561 = 726
726+627 = 1353
1353+3531 = 4884 --> 4
```

2. Construya la función `int compara(char *,char *)`, la cual permite comparar 2 strings lexicográficamente. Los valores de retorno deben ser los mismos que los entregados por la función `strcmp` perteneciente al lenguaje C.
3. Desarrolle dos funciones en lenguaje C que permitan encontrar el valor de π mediante la siguiente expresión:

$$\pi = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \dots$$

- (a) (10 pts.) Una función considera que la precisión del cálculo depende de la cantidad de términos de la multiplicatoria: se consideran los n primeros términos, donde n es un parámetro de la función.
- (b) (15 pts.) La otra función considera que la precisión del cálculo depende de la diferencia entre dos aproximaciones consecutivas del valor de π : si esta es menor que el parámetro de la función ϵ se detiene el cálculo.

Ayuda: el i -ésimo término se puede generar usando i . Por ejemplo el **cuarto** término se puede generar usando $\frac{4}{4+1}$.

4. Considere que el arreglo bidimensional donde se almacena el texto es definido de la sgte manera: **char texto[10][100]**; y las palabras se encuentran separadas por un espacio.

Escriba la función

int diferentes(char texto[][numcols],int numFils,int numCols); que retorna la cantidad de palabras diferentes dentro del texto.

5. Un número de nueve dígitos se dice deleitable si:
- (a) contiene exactamente los dígitos entre 1 y 9, una vez cada uno, y
 - (b) los números creados tomando los n primeros dígitos ($1 \leq n \leq 9$) son cada uno de ellos divisibles por n , de tal forma que el primer dígito será divisible por 1 (siempre lo será), los dos primeros dígitos forman un número divisible por 2, los tres primeros forman un número divisible por 3 y así sucesivamente. Por ejemplo: si consideramos el número 123456789.

$$\begin{aligned}1:1 &= 1 \\12:2 &= 6 \\123:3 &= 41\end{aligned}$$

sin embargo 1234 no es divisible por 4.

Escriba la función **int deleitable(long int)** que permite determinar si un número es o no deleitable.

4 Ruteos

Realice el ruteo de los sgtes programas mostrando el contenido de las variables dentro de cada función y la salida.

- 1.
- ```
int n=76;
int f(int *);

main(){
 int i=500;
 char c;

 c = f(&i);
 printf(“%d %d %d”,c,i,n);
}

int f(int *q){
 *q = ++n + *q;
 return *q;
}
```

2.

```
int r=9;
int f(int);

main(){
 int j=5,r;

 r = f(j);
 printf(''%d'',r,j);
}
int f(int x){
 x = x + r;
 return x;
}
```

3.

```
main(){
 int a, f=4,z;

 a = 'c'-1 == 'a';
 z = a * 2 + f%2 * 2;
 printf(''%d %d'',a,z);
}
```

4.

```
#define n 2
main(){
 int a[n][n]={1,2},{3,4}};
 int r;
 printf(''%d'', f(a));
}

int f(a[][n]){
 int i,j,s=0;

 for(i=0;i<=n-1;i++)
 for(j=0;j<=n-1;j++)
 s+=a[i][j];
 return s;
}
```

## 5 Listas Enlazadas

1. Considere dos listas enlazadas a y b cuyo contenido se encuentra ordenado en forma ascendente. Escriba una función en lenguaje C que genere una tercera lista que contenga la intersección de a y b. La función a desarrollar debe tener el sgte prototipo:

**struct nodo \* interseccion(struct nodo \*,struct nodo \*);**

2. En un archivo de texto llamado "datos.txt" se encuentran almacenados número enteros (uno por línea). Escriba un programa en C que genere una lista enlazada a partir del contenido del archivo. Una vez creada la lista escriba una función que elimine los elementos repetidos. Considere el uso de funciones con el sgte prototipo:

**void cargarLista(struct nodo\*\*,char \*);**

**void eliminarRepetidos(struct nodo \*\*);**

3. Escriba una función que determine si el contenido de una lista enlazada es un palíndrome.

**int testPalindrome(struct nodo \*);**

4. Escriba una función que elimine todos los nodos de una lista en forma alternada.

**void eliminarLista(struct nodo \*\*);**

5. Escriba una función que determine si dos listas enlazadas A y B se encuentran unidas (en forma de Y).

**int estanUnidas(struct nodo \*, struct nodo \*);**

6. Escriba una función que determine el n-ésimo nodo de una lista a partir desde el último nodo de ésta. Considere que la lista se encuentra apuntada sólo por el puntero inicio.

**struct node \* nthNodo(struct nodo \*);**

7. Se tiene un puntero p que apunta a un nodo de una lista. Escriba una función que elimine el nodo apuntado por p. Considere que no se conoce el inicio de la lista.

**void eliminarNodo(struct nodo \*);**

8. Considere dos números enteros almacenados dígito por dígito en dos listas enlazadas apuntadas por n1 y n2 respectivamente. Escriba una función que genere una 3ra lista apuntada por n3 que contiene en cada uno de sus nodos los dígitos del resultado de la suma de n1 y n2.

**struct nodo \* suma(struct nodo \*, struct nodo \*);**

9. Se tiene 2 listas enlazadas apuntadas por a y b respectivamente cuyo contenido se encuentra ordenado en forma ascendente. Escriba una función que genere la mezcla de ambas listas (apuntada por c) en orden ascendente sin crear nuevos nodos.

**struct nodo \* mezclar(struct nodo \*, struct nodo \*);**