

ICI-142 Fundamentos de Programación

Guía de Ejercicios Preparación Certamen #1

Makarena Donoso <makarena.donos@gmail.com>

Wenceslao Palma <wenceslao.palma@ucv.cl>

1 Funciones, arreglos unidimensionales y bidimensionales

1. 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 (basado en funciones) 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.

2. 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 una función en lenguaje C, que permita determinar la raíz cuadrada de un número en base a la recurrencia anterior.

3. 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.
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. Desarrolle un programa en lenguaje C que permita rotar una matriz en ángulos de 0,90,180 ó 270 grados y además comparar la matriz rotada con la original. El programa debe: ingresar valores en una matriz de NxN y el ángulo en que se desea rotar, mostrar la matriz rotada en el ángulo que corresponda y calcular la cantidad de elementos de la matriz original que son mayores que los elementos de la matriz rotada, considerando los elementos en posiciones equivalentes. Por ejemplo:

```

1 2   rotación en 90 grados 2 4
3 4                               1 3

```

elementos mayores: 2 (3>1 y 4>3)

6. Considere dos cadenas (arreglos de caracteres) llamadas A y SEPARADORES, cuyos contenidos son ingresados desde teclado. El programa a escribir debe ser tal que si :

- (a) La cadena A contiene: “sin embargo, es la dinámica autopoietica la responsable de permitir la fractura en un plano permitido . . . fenómeno muy particular en....”
- (b) Y los SEPARADORES son : “ , . ”

Debe salir por pantalla:

- (1) sin embargo
- (2) es la dinámica autopoietica la responsable de permitir la fractura en un plano permitido
- (3) fenómeno muy particular en

7. Escriba un programa en base a funciones que dado un mensaje calcule su costo para enviarlo por telégrafo. Para esto se sabe que las letras cuestan, cada una, \$10. Los caracteres especiales que no sean letras cuestan \$30 y los dígitos tienen un valor de \$20 cada uno. Los espacios no tienen valor. Restricciones :

- El mensaje es una cadena
- Las letras ñ,á,é,í,ó,ú se consideran caracteres especiales.

Un ejemplo de ejecución del programa es :

Entrada : Feliz cumpleaños

Salida : Valor del mensaje : \$ 170

8. Escriba una función 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:

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

2 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;
}
```