

Clase Práctica

Ejercicio 1: Para las siguientes situaciones, realice un módulo recursivo y la traza con los datos indicados para cada una:

- Desarrollar un módulo recursivo que permita sumar los dígitos de un número natural X. Realizar la traza para $X = 12345$.
- Desarrollar un módulo recursivo que permita determinar si un dígito D pertenece a un número entero positivo N. Realizar la traza para $D = 1$ y $X = 45673$.
- Desarrollar un modulo recursivorecursiva en C que calcula el término de la serie de Fibonacci para un índice dado. Realiza la traza para $N=8$

Ejercicio 2: Desarrolla la traza del siguiente codigo en C que imprime los números primos menores que un número natural N. Realizar la traza para $N = 100$, luego con $N = 27$.

```
#include <stdio.h>

int esprimo(int n) {
    int i;
    for (i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            return 0;
        }
    }
    return 1;
}

void primos(int n) {
    if (n == 1) {
        return;
    }
    if (esprimo(n)) {
        printf("%d\n", n);
    }
    primos(n - 1);
}

int main() {
    int n;
    printf("Ingrese un numero natural: ");
    scanf("%d", &n);
    printf("Los numeros primos menores que %d son: \n", n);
    primos(n);
    return 0;
}
```

Ejercicio 3: Se pide crear un programa recursivo que permita generar aleatoriamente una lista de N números enteros en el intervalo [A,B], posteriormente, ingresar un número determinar si es divisor de algunos de los numeros recientemente generados. Mostrar los numeros en los cuales es divisor.

Ejercicio 4: Definir función recursiva llamada numerologia, que devolverá un numero a partir de una fecha de nacimiento. El número se calcula sumando cada uno de los dígitos de la fecha completa: día, mes y año. Por ejemplo, si alguien nació el 26 de marzo de 1966, sumará: $2+6+0+3+1+9+6+6=33$, se continúa reduciendo $3+3=6$, el programa deberá devolver el numero 6 en este caso. Pautas: La fecha se ingresara con el siguiente formato : 25/05/1979. Deberá haber una funcion recursiva que sume.

Ejercicios Resueltos

Ejercicio 1: Para las siguientes situaciones, realice un módulo recursivo y la traza con los datos indicados para cada una:

a.- Desarrollar un módulo recursivo que permita sumar los dígitos de un número natural X. Realizar la traza para $X = 12345$.

```
#include <stdio.h>

int sumar_digitos(int x);

int sumar_digitos(int x) {
    if (x == 0) {
        return 0;
    } else {
        return x % 10 + sumar_digitos(x / 10);
    }
}

int main() {
    printf("%d\n", sumar_digitos(12345));
    return 0;
}
```

b.- Desarrollar un módulo recursivo que permita determinar si un dígito D pertenece a un número entero positivo N. Realizar la traza para $D = 1$ y $X = 45673$.

```
#include <stdio.h>

int contieneDRecv(int arr[], int index, int size, int D, int *numConDigito) {
    if (index == size) {
        return 0;
    }

    int num = arr[index];
    int original = num;
    while (num > 0) {
        if (num % 10 == D) {
            *numConDigito = original;
            return 1;
        }
        num /= 10;
    }

    return contieneDRecv(arr, index + 1, size, D, numConDigito);
}

int main() {
    int numeros[4] = {456, 73, 821, 915};
    int D = 1;
    int numConDigito = 0;
    int size = sizeof(numeros) / sizeof(numeros[0]);
```

```
        if (contieneDRecv(numeros, 0, size, D, &numConDigito)) {  
            printf("%d esta presente en el numero %d del arreglo.\n", D, numConD  
        } else {  
            printf("%d no esta presente en ninguno de los numeros del arreglo.\n"  
        }  
  
        return 0;  
    }  
}
```

c.-Desarrollar un modulo recursivorecursiva en C que calcula el término de la serie de Fibonacci para un índice dado. Realiza la traza para N=8.

```
#include <stdio.h>  
  
int fib(int n);  
  
int main() {  
    int n = 8;  
    printf("El termino %d de la serie de Fibonacci es: %d\n", n, fib(n));  
    return 0;  
}  
  
int fib(int n) {  
    if (n == 0) {  
        return 0;  
    } else if (n == 1) {  
        return 1;  
    } else {  
        return fib(n - 1) + fib(n - 2);  
    }  
}
```

Ejercicio 3: Para las siguientes situaciones, realice un módulo recursivo y la traza con los datos indicados para cada una:

```
#include <stdio.h>  
  
int esprimo(int n) {  
    int i;  
    for (i = 2; i * i <= n; i++) {  
        if (n % i == 0) {  
            return 0;  
        }  
    }  
    return 1;  
}  
  
void primos(int n) {  
    if (n == 1) {  
        return;  
    }  
    if (esprimo(n)) {  
        printf("%d\n", n);  
    }  
    primos(n - 1);  
}
```

```
    }
    primos(n - 1);
}

int main() {
    int n;
    printf("Ingrese un numero natural: ");
    scanf("%d", &n);
    printf("Los numeros primos menores que %d son: \n", n);
    primos(n);
    return 0;
}
```

Ejercicio 4: Definir función recursiva llamada numerologia, que devolverá un numero a partir de una fecha de nacimiento. El número se calcula sumando cada uno de los dígitos de la fecha completa: día, mes y año. Por ejemplo, si alguien nació el 26 de marzo de 1966, sumará: $2+6+0+3+1+9+6+6=33$, se continúa reduciendo $3+3=6$, el programa deberá devolver el numero 6 en este caso. Pautas: La fecha se ingresara con el siguiente formato : 25/05/1979. Deberá haber una funcion recursiva que sume.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int sumar_digitos(char *num, int suma);

void numerologia() {
    char fecha[20];
    printf("Ingrese su fecha de nacimiento con el siguiente formato dd/mm/aaaa:");
    scanf("%s", fecha);
    char *numeros = strtok(fecha, "/");
    int suma = 0;

    while (numeros != NULL) {
        suma = sumar_digitos(numeros, suma);
        numeros = strtok(NULL, "/");
    }

    printf("%d\n", suma);
}

int sumar_digitos(char *num, int suma) {
    for (int i = 0; num[i] != '\0'; i++) {
        if (num[i] != '0') {
            suma += num[i] - '0';
        }
    }

    if (suma > 10) {
        char temp[20];
        sprintf(temp, "%d", suma);
    }
}
```

```
        suma = sumar_digitos(temp, 0);  
        return suma;  
    } else {  
        return suma;  
    }  
}  
  
int main() {  
    numerologia();  
    return 0;  
}
```