

UNIVERSIDAD DE LIMA

Facultad de Ingeniería y Arquitectura

Carrera de Ingeniería de Sistemas

Sistemas Operativos

Laboratorio 04

Programación en C (I)



Septiembre 2022



Objetivos

1. Programar formatos de entrada y salida, cadenas, bucles (loops), arrays, funciones y estructuras en lenguaje C
2. Analizar la programación, desarrollo, funcionamiento y resultados de los algoritmos planteados

01

Introducción

El lenguaje de programación “C” fue desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell para ser usado en UNIX. Es un lenguaje de programación de tipo estructurado, similar a Pascal, Fortran, Basic. “C” es bastante eficiente en los resultados y es el lenguaje de programación más utilizado en la creación de software de sistemas, siendo también utilizado para crear aplicaciones.

C utiliza instrucciones y sentencias como if, else, for, do y while. A pesar de ser un lenguaje de alto nivel (puesto que es estructurado y posee sentencias y funciones que simplifican su funcionamiento), se puede programar a bajo nivel (similar a lenguaje ensamblador).

02

Formatos de entrada y salida

Para los ejercicios de programación planteados, escriba el programa en C, compile, ejecute y analice los resultados.

La función *printf*

La función **printf (print formatted)** es utilizada para mostrar una cadena de caracteres con formato. Esta función permite dar formato específico a la salida, siguen algunos ejemplos:

%d ó %i	entero en base 10 con signo (int)
%u	entero en base 10 sin signo (int)
%o	entero en base 8 sin signo (int)
%x	entero en base 16, letras en minúscula (int)
%X	entero en base 16, letras en mayúscula (int)

Ejemplo 01: Escriba el programa en C, compile, ejecute y analice los resultados:

```
#include <stdio.h>

int main(void)
{
    int i;
    float x;

    i= 40;
    x= 839.21f;

    printf("|%d|%5d|%-5d|%5.3d|\n", i, i, i, i);
    printf("|%10.3f|%10.3e|%-10g|\n", x, x, x);

    return(0);
}
```

La función *scanf*

La función **scanf (scan-format)** analiza una entrada de datos con formato y cargan el resultado en argumentos.

Ejemplo 02: Escriba el programa en C, compile, ejecute y analice los resultados.

```
#include <stdio.h>

int main(void)
{
    int num1, denom1, num2, denom2, result_num, result_denom;

    printf("Ingrese primera fracción:");
    scanf("%d/%d", &num1, &denom1);

    printf("Ingrese segunda fracción:");
    scanf("%d/%d", &num2, &denom2);

    result_num = num1 * denom2 + num2 * denom1;

    result_denom = denom1 * denom2;
    printf("La suma es %d/%d\n", result_num, result_denom);

    return(0);
}
```

03

Revisión de bucles

Bucle While

Condición simple que al evaluarse devuelve el valor verdadero o falso, se repite mientras la condición sea verdadera, cuando es falsa, el programa pasa a la instrucción siguiente.

Ejemplo 03: Escriba los programas en C, compile, ejecute y analice los resultados.

```
#include <stdio.h>

int main(void)
{
    int i, n;

    printf("Este programa imprime una tabla de cuadrados.\n");
    printf("Ingrese el numero de entradas para la tabla:");
    scanf("%d", &n);

    i=1;
    while (i<=n)
    {
        printf("%10d%10d\n", i, i*i);
        i++;
    }
    return(0);
}
```

```
#include <stdio.h>

int main(void)
{
    int n, sum=0;

    printf("Este programa suma una serie de enteros.\n");
    printf("Ingrese enteros (ponga 0 para terminar):");

    scanf("%d", &n);
    while (n != 0){
        sum += n;
        scanf("%d", &n);
    }
    printf("La suma es: %d\n", sum);

    return(0);
}
```

Bucle Do

En un bucle **do while**, primero se ejecuta el bloque de instrucciones y, después, se evalúa la condición. En el caso de que ésta sea verdadera, se vuelve a ejecutar el bloque de instrucciones, y así sucesivamente, hasta que, la condición sea falsa.

Ejemplo 04: Escriba el programa en C, compile, ejecute y analice los resultados.

```
#include <stdio.h>
int main(void)
{
    int digits = 0, n;

    printf("Ingrese un entero no negativo:");
    scanf("%d", &n);

    do{
        n/=10;
        digits++;
    } while (n > 0);

    printf("El numero tiene %d digitos.\n", digits);

    return(0);
}
```

Bucle For

El bucle for es ideal usarlo cuando se sabe el número de veces (iteraciones) que tiene que ejecutarse un determinado bloque de instrucciones.

Ejemplo 05: Escriba el programa en C, compile, ejecute y analice los resultados.

```
#include <stdio.h>
int main(void)
{
    int i, n;

    printf("Este programa imprime una tabla de cuadrados.\n");
    printf("Ingrese el numero de entradas para la tabla:");
    scanf("%d", &n);

    for (i=1;i <= n; i++)
        printf("%10d%10d\n",i,i*i);

    return(0);
}
```

Ejemplo 06: Se presenta el código que permite convertir un número entre 0 a 255 a base 2:

```
#include <stdio.h>

int main()
{
    int n_decimal ;
    int tope = 128 ;

    printf("*Digite numero: ") ;
    scanf("%d",&n_decimal) ;

    printf("%d = ",n_decimal) ;

    for(int i=0;i<=7;i++)
    {
        if (n_decimal>=tope)
        {
            n_decimal -= tope ;
            printf("1") ;
        }
        else printf("0") ;
        tope /= 2 ;
    }
}
```

Ejemplo 07: Modificar el código anterior de tal manera que se implemente una función que permita leer números enteros y que una vez ingresado se detecte si se encuentra en el rango [0,255]

Ejemplo 08: Codificar un programa en C que muestre un menú conteniendo opciones para efectuar las cuatro operaciones aritméticas básicas con números enteros y una opción adicional para salir del programa. Una vez seleccionada la opción, implementar la opción aritmética correspondiente para cada opción seleccionada.

```
#include <ctype.h>
#include <stdlib.h>

int main()
{
    char tecla ;
    int a, b ;
```

```

do
{
    system("clear") ;
    printf("1. Suma\n") ;
    printf("2. Resta\n") ;
    printf("3. Multiplicacion\n") ;
    printf("4. Division\n") ;
    printf("5. Salir\n") ;
    printf("\nOpcion: ") ;
    tecla = getchar() ;
}

while(tecla!='1'&&tecla!='2'&&tecla!='3'&&tecla!='4'&&tecla!='5') ;

switch(tecla)
{
    case '1': printf("Operacion: Suma\n") ;
              printf("a = ") ; scanf("%d",&a) ;
              printf("b = ") ; scanf("%d",&b) ;
              printf("r = %d",a+b) ;
              break ;
    case '2': printf("Operacion: Resta\n") ;
              printf("a = ") ; scanf("%d",&a) ;
              printf("b = ") ; scanf("%d",&b) ;
              printf("r = %d",a-b) ;
              break ;
    case '3': printf("Operacion: Multiplicacion\n") ;
              printf("a = ") ; scanf("%d",&a) ;
              printf("b = ") ; scanf("%d",&b) ;
              printf("r = %d",a*b) ;
              break ;
    case '4': printf("Operacion: División\n") ;
              printf("a = ") ; scanf("%d",&a) ;
              printf("b = ") ; scanf("%d",&b) ;
              printf("r = %d",a/b) ;
              break ;
}
printf("\nSalida ...\n") ;

return 0 ;
}

```

Ejemplo 10: Generar 10 números aleatorios enteros entre 0 y 100 y mostrarlos en pantalla.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 10

int main()
{
    int numero ;

    for(int i=0;i<MAX;i++)
    {
        numero = rand() % 101 ;
        printf("n[%d] = %d\n", i, numero) ;
    }

    return 0 ;
}
```

05

Ejercicios propuestos

- Ejercicio 01:** Obtener la factorial de un número entero utilizando funciones recursivas
- Ejercicio 02:** Generar 30 calificaciones de examen parcial y mostrar la nota máxima, la nota mínima y el promedio con dos dígitos de precisión
- Ejercicio 03:** Codificar un programa en Lenguaje C que solicite el año, el mes y que muestre en pantalla el número de días que tiene dicho mes.
- Ejercicio 04:** Proponer un algoritmo y luego codificar un programa en Lenguaje C que muestre en pantalla todos los números enteros primos entre 1 y n, donde n debe ser mayor que 20 y menor que 200
- Ejercicio 05:** Proponer un algoritmo y luego codificar en lenguaje C un programa que muestre por pantalla el área de los círculos de radios 1, 2, 3, 4 y 5