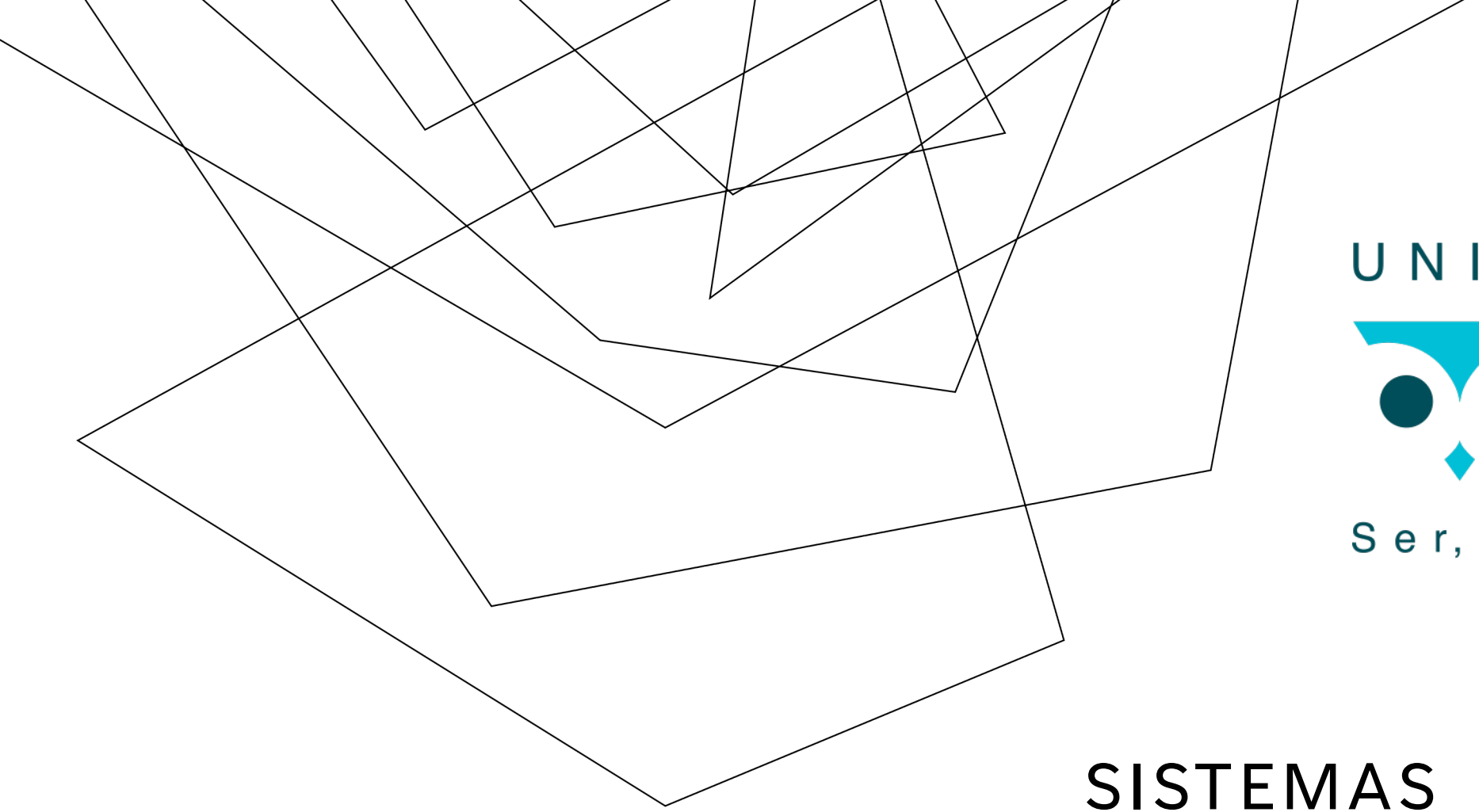


UNIVERSIDAD EIA®

Ser, Saber y Servir

SISTEMAS OPERATIVOS

Semestre 2022-2

INSTALACIÓN C WIN

https://sourceforge.net/projects/mingw/files/

https://code.visualstudio.com/

- Instalamos vscode
- Instalamos el compilador mingw
- Agregamos en las variables de entorno la ruta C:\MinGW\bin (o la ruta que usaste al instalar)
    - Variables de usuario editar Path y en este hacemos una nueva variable de entorno con la ruta anteriormente mencionada
- Abrir cmd y ejecutar gcc --version
- Abrimos vscode -> settings -> user -> extensions -> run code configuration -> ENABLE run in terminal
- Luego instalamos las extenciones de c/c++ de windows y la de code runner de jun han en vscode

UNIVERSIDAD EIA®

Ser, Saber y Servir

C ONLINE

https://replit.com/languages/c

UNIVERSIDAD EIA®

Ser, Saber y Servir

PARADIGMAS DE PROGRAMACIÓN

- Paradigma imperativo
  - Programación orientada a objetos
  - Programación procedimental
  - Programación estructurada
- Paradigma declarativo
  - Programación funcional
  - Programación lógica

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- 1972 – Dennis Ritchie
- Lenguaje B
- Instrucciones tanto de nivel bajo como de nivel alto
- Lenguaje de Medio nivel, compilado (analiza, genera archivo binario y ejecuta), de propósito general, estructurado (ejecuta línea a línea) y modular, altamente portable, muy eficiente pero no permite funciones anidadas, problemas con la precedencia de operadores, nativamente no tiene programación multihilo (se necesitan librerías).

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Lenguaje tipado
- Apuntadores – son variables que contienen la dirección de una variable
- Usado en
  - Software de aplicación
  - Drivers
  - Sistemas operativos
  - Supercomputadoras
  - Sistemas embebidos
- https://www.youtube.com/watch?v=CYvJPra7Ebk&t=13s

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Que no encontraremos en C (nativo)
  - Hilos
  - Clases
  - Objetos
  - Recolección de basura
    - Funciones
      - malloc
      - free

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Comentarios
  - Una línea //
  - Varias líneas /* */
- Siempre cerramos con ;
- #Include – para invocar librerías
- <stdio.h> - librería base para funciones de entrada, salida y manipulación de ficheros
- Main – método base
- Una variable siempre debe empezar con una letra o _
- Se entiende que 0 es falso, cualquier otro valor es verdadero
- Statement – línea de código -> printf("Hola");
- Bloque de codigos – es un conjunto de statements que se agrupan por medio de {}
- Shift+Alt+F para formatear el codigo

UNIVERSIDAD EIA®

Ser, Saber y Servir

HOLA MUNDO

Vamos a crear una carpeta y en esta crear un archivo llamado hello.c

```c
#include <stdio.h>

int main()
{
    printf("hello");
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

IMPRIMIR EN PANTALLA

- printf()

%d	int
%f	float
%lf	long float
%c	char
\n	Salto de linea

UNIVERSIDAD EIA®

Ser, Saber y Servir

TIPOS DE VARIABLES

Enteros	Int unsigned int short unsigned short long unsigned long	4 bytes 4 bytes 2 bytes 2 bytes 8 bytes 8 bytes	-2147483648 a 2147483647 0 a 4294967295 -32768 a 32767 O a 65535 -9223372036854775808 a 9223372036854775807 0 a 18446744073709551615
Punto flotante	float double long double	4 bytes 8 bytes 10 bytes	1,2e-38 a 3.4e+38 2,3e-308 a 1,7e+308 3,4e-4932 a 1,1e+4932
Caracteres	char unsigned char	1 byte 1 byte	-128 a 127 0 a 255
Vacio	void		

VARIABLES

- Declaración
  - int num;
- Asignación
  - int num= 1;

- Variables locales
  - Declaración dentro de la función main
- Variables globales
  - Declaración por fuera del main
- Variables externas
  - Declaración por fuera del main y con la palabra reservada extern al inicio de la declaración, igual también debemos declararlas en cada función que vayamos en la que vayamos a utilizarlas

UNIVERSIDAD EIA®

Ser, Saber y Servir

ENTEROS

```c
#include <stdio.h>

int main()
{
    int num1, num2;
    int num3 = 3;
    num1 = 2;
    printf("num3 es %d" , num3);
    printf ("num2 es %d y num1 es %d" ,
    num2, num1);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

PUNTO FLOTANTE

```c
#include <stdio.h>

int main()
{
    float pi = 3.1416;
    printf ("el valor de pi es %f", pi);
    printf ("el valor de pi es %.2f ", pi);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

CHAR

```
#include <stdio.h>

int main()
{
    char char1 = 'b';
    char char2 = 110;
    char char3[15] = “juan”;
    char char4[15];
    scanf(”%s ", &char4);
    printf("el valor del char1 es %c",
    char1);
    printf("el valor del char2 es %c",
    char2);
    printf("el valor del char3 es %s",
    char3);
    printf("el valor del char4 es %s",
    char4);

    return 0;
}
```

Tabla ASCII
https://upload.wikimedia.org/wikipedia/commons/1/1b/ASCII-Table-wide.svg

UNIVERSIDAD EIA®
Ser, Saber y Servir

CONSTANTES

entero	const int num = 223;
float	const float numf = 2.345;
char	const char charcons = ‘a’;

- Directivas de preprocesador
  - #define num 8 -> nunca se podrá cambiar

UNIVERSIDAD EIA®

Ser, Saber y Servir

OPERACIONES ARITMÉTICAS

+	Suma
-	Resta
*	Multiplicación
/	División - cociente
%	División – resto - modulo

UNIVERSIDAD EIA®

Ser, Saber y Servir

OPERACIONES LÓGICAS

Relación	
Mayor	>
Menor	<
Igual	==
Mayor que	>=
Menor que	<=
Distinto de	!=

Lógicos	
And	&&
Or	||
Not	!

UNIVERSIDAD EIA®

Ser, Saber y Servir

IF Y IF ANIDADO

```c
#include <stdio.h>

int main()
{
    int var = 3;

    if (var == 1)
        printf("el numero es 1");
    else if (var == 2)
        printf("el numero es 2");
    else if (var == 3)
        {
                printf ("el numero es 3 \n");
                printf("esto es un bloque");
        }
    else
        printf("no es ni 1 ni 2 ni 3");
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

SWITCH Y SWITCH ANIDADO

```c
#include <stdio.h>

int main()
{
    int val = 3;

    switch (val)
    {
        case 0: printf("caso 1"); break;
        case 1: printf("caso 2"); break;
        default: printf("default"); break;
    }
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

WHILE AND DO WHILE

```c
#include <stdio.h>

int main()
{
    int val = 3;

    while (val < 5)
    {
        val++;
        printf("entre en un while");
    }

    do{
        printf("entre al do");
    } while (val < 5);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

FOR

```c
#include <stdio.h>

int main()
{
    for (int count = 1; count <= 5; count++)
    {
        printf("estoy en el for \n");
    }

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

BREAK, CONTINUE, GOTO

```c
#include <stdio.h>

int main()
{
    int acum = 0;
    int count;
    for (count = 0; count <= 5; count++)
    {
        continue;
        printf("estoy en el for \n");
        acum = acum + 1;
    }

    printf("%d \n", count);
    printf("%d \n", acum);

    return 0;
}
```

Continue: cuando quieres seguir corriendo un ciclo pero por algún condicional quieres hacer skip de una iteración en específico

Goto: con esta instrucción controlamos errores

UNIVERSIDAD EIA®

Ser, Saber y Servir

BREAK, CONTINUE, GOTO

```c
#include <stdio.h>

int main()
{
    int i;
    for (i = 1; i <= 5; ++i)
    {
        printf("estoy en el for \n");
        if (i == 3)
        {
                goto jump;
        }
    }

    jump:
        printf("Salte");

    return 0;
}
```

Continue: cuando quieres seguir corriendo un ciclo pero por algún condicional quieres hacer skip de una iteración en específico

Goto: con esta instrucción controlamos errores

UNIVERSIDAD EIA®

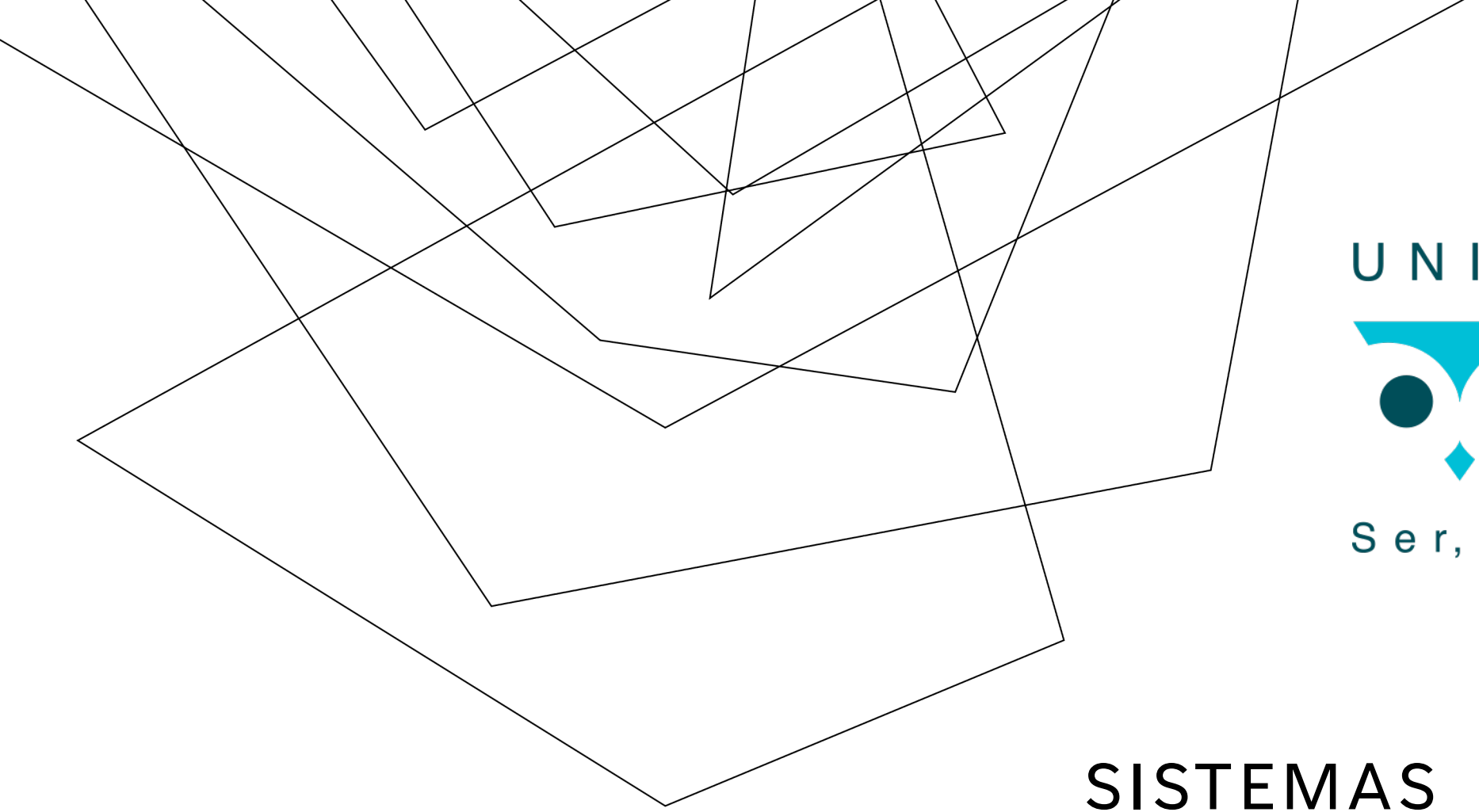
Ser, Saber y Servir

FUNCIONES

```c
#include <stdio.h>

int func (int n);

int main()
{
    for (int count = 1; count <= 5; count++)
    {
        printf("estoy en el for y traigo de la funcion
        func el numero %d \n", func(count));
    }
    return 0;
}

int func (int n)
{
    printf("acabe de entrar a la funcion ");
    n++;
    return n;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir



UNIVERSIDAD EIA®

Ser, Saber y Servir

SISTEMAS OPERATIVOS

Semestre 2022-2

TYPEDEF

```c
#include <stdio.h>

int main(){

    typedef int Entero;

    Entero num1 = 54, num2 = 75, sum = num1
    + num2;

    printf("el valor de la suma es de %i",
    sum);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ASIGNACION

```c
#include <stdio.h>

int main(){
    int a,b,c;
    a=b=c=20;

    printf("el valor de a es = %i \n", a);
    printf("el valor de b es = %i \n", b);
    printf("el valor de c es = %i \n", c);

    return 0;
}
```

¿Que pasaria si declaramos `int a,b,c=20;` ?

`+= -= *= /= %=`

UNIVERSIDAD EIA®

Ser, Saber y Servir

FUNCIÓN SCANF Y GETS

```c
#include <stdio.h>

int main(){
    char e[40];
    printf("digite el valor \n");
    scanf("%s",e);
    printf("el valor de la variable es %s",e);

    char f[40];
    printf("digite el valor \n");
    gets(f);
    printf("el valor de la variable es %s",f);
    return 0;

    float num1,num2,suma;
    printf("Esribe 2 numeros \n");
    scanf("%f %f",&num1,&num2);
    suma = num1 + num2;
    printf("la suma da un igual de %.2f", suma);

}
```

Scanf apunta a la memoria en
todos los tipos de datos excepto el char
en arreglo - &var

Si queremos evitar inyeccion de data en la
captura, podemos usar fgets(f,20,stdin);

Fflush(stdin); limpia el buffer se utiliza
por lo general antes de la captura de info

UNIVERSIDAD EIA®

Ser, Saber y Servir

OPERADOR ?

```c
#include <stdio.h>

int main(){
    int num;
    printf("Escribe numero \n");
    scanf("%i",&num);

    (num%2==0) ? printf("Es par \n") : printf("Es impar \n");

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

LIBRERÍA STRING.H
STRCPY

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

int main(){
    char orig[] = "Copia";
    char dest[12];

    strcpy(dest,orig);

    printf("%s",dest);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

LIBRERÍA STRING.H
STRCAT

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

int main(){
    char nom1[] = "Juan", nom2[] = "Guillermo", concat[41];

    strcat(concat,nom1);
    strcat(concat,"-");
    strcat(concat,nom2);
    printf("%s",concat);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

LIBRERÍA STRING.H
STRCMP

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

int main(){
    char nom1[] = "juan";

    char nomcomp[20];

    printf("escriba el nombre a comparar: ");
    fflush(stdin);
    scanf("%s", nomcomp);

    if (strcmp(nom1,nomcomp) == 0){
        printf("son iguales");
    }else{
        printf("no son iguales");
    }

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

LIBRERÍA STRING.H
STRLEN

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

int main(){

    char nomcomp[20];

    printf("escriba el nombre: ");
    fflush(stdin);
    scanf("%s", nomcomp);
    printf("\n el nombre tiene %lu caracteres",strlen(nomcomp));

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

LIBRERÍA STRING.H
STRRCHR

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

int main(){

    char abc[] = "abcdefghijklmnopqrstuvwxyz";
    char inicial;

    printf("escriba la letra desde la que desea iniciar: ");
    scanf("%c",&inicial);
    printf("\n%s",strrchr(abc,inicial));

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARREGLOS
LISTAS

```c
#include <stdio.h>

int main(){

    int arreglo[3] = {24,77,54};

    printf("%i\n",arreglo[0]);
    printf("%i\n",arreglo[1]);
    printf("%i\n",arreglo[2]);

    arreglo[0] = 85;
    printf("%i\n",arreglo[0]);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARREGLOS
MATRICES

```c
#include <stdio.h>

int main(){

    int arreglo[3][3] = {{24,77,54},{55,64,87},{15,14,96}};
    //filas - columnas
    printf("%i\n",arreglo[0][0]);
    printf("%i\n",arreglo[1][0]);
    printf("%i\n",arreglo[2][0]);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ESTRUCTURAS

```c
#include <stdio.h>

struct alumno{
    char nombre[20];
    char carrera[30];
    float nota;
}
alumno1 = {"camilo", "sistemas", 2.4},
alumno2 = {"andres", "sistemas", 1.5};

int main(){

    printf("su nombre es: %s \n", alumno1.nombre);
    printf("su carrera es: %s \n", alumno1.carrera);
    printf("su edad es: %.2f \n\n", alumno1.nota);

    printf("su nombre es: %s \n", alumno2.nombre);
    printf("su carrera es: %s \n", alumno2.carrera);
    printf("su edad es: %.2f \n", alumno2.nota);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ESTRUCTURAS

```c
#include <stdio.h>

struct alumno{
    char nombre[20];
    char carrera[30];
    float nota;
}alumno1;

int main(){

    printf("escriba el nombre: ");
    gets(alumno1.nombre);
    printf("escriba carrera: ");
    gets(alumno1.carrera);
    printf("escriba nota: ");
    scanf("%f",&alumno1.nota);

    printf("\nsu nombre es: %s \n", alumno1.nombre);
    printf("su carrera es: %s \n", alumno1.carrera);
    printf("su edad es: %.2f \n\n", alumno1.nota);

    return 0;
}
```

Las estructuras tambien se pueden usar como arreglos de estructuras – alumno[5]

UNIVERSIDAD EIA®

Ser, Saber y Servir

ESTRUCTURAS ANIDADAS

```c
#include <stdio.h>

struct notas
{
    float nota;
};
struct alumno{
    char nombre[20];
    char carrera[30];
    struct notas notaparcial;
}alumno[2];

int main(){

    printf("escriba el nombre: ");
    gets(alumno[0].nombre);
    printf("escriba carrera: ");
    gets(alumno[0].carrera);
    printf("escriba nota: ");
    scanf("%f",&alumno[0].notaparcial.nota);

    printf("\nsu nombre es: %s \n", alumno[0].carrera);
    printf("su carrera es: %s \n", alumno[0].carrera);
    printf("su edad es: %.2f \n\n", alumno[0].notaparcial.nota);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

APUNTADORES

```c
#include <stdio.h>

int main(){

    int num = 2;
    printf("el valor de la variable es: %i",num);
    printf("\nla direccion en memoria es: %p",&num);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

APUNTADORES

```c
#include <stdio.h>

int main(){

    int num = 2;
    int *pNum;

    printf("el valor de la variable es: %i",num);
    printf("\nla direccion en memoria es: %p",&num);

    pNum = &num;
    printf("\nel valor de la variable es: %i",num);
    printf("\nel valor de la variable es: %i",*pNum);

    printf("\nla direccion en memoria es: %p",&num);
    printf("\nla direccion en memoria es: %p",pNum);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

APUNTADORES

```c
#include <stdio.h>

void apunt(int *a){
    *a += 5;
}

int main(){

    int num;
    printf("escriba un numero: ");
    scanf("%i", &num);
    printf("\nel numero antes de la funcion es: %i", num);
    apunt(&num);
    printf("\nel numero despues de la funcion es: %i", num);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

APUNTADORES

```c
#include <stdio.h>

void apunt(int *a){
    a += 2;
    *a = 5;
}

int main(){

    int num[5] = {1,2,3,4,5};
    printf("\nel valor de la posicion 3 antes de la funcion es: %i",
    num[2]);
    apunt(&num[0]);
    printf("\nel numero despues de la funcion es: %i", num[2]);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARCHIVOS

r	Lectura, el archivo debe existir
w	Escribe documento, si no existe lo crea, si existe lo sobrescribe
a	Escribe al final del documento, si no existe se crea
r+	Lectura y escritura (el archivo debe existir)
w+	Crea archivo, si este existe lo sobrescribe

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARCHIVOS

```c
#include <stdio.h>

int main(){

    FILE *fd;

    char direccion[] = "file.txt";

    fd = fopen(direccion,”r");

    if (fd == NULL){
        printf("archivo no existe");
    } else{
        printf("se encontro el archivo en la direccion: %s", direccion);
    }

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARCHIVOS

```c
#include <stdio.h>

int main(){

    FILE *fd;

    char direccion[] = "file.txt";

    fd = fopen(direccion,"a");

    if (fd == NULL){
        printf("archivo no existe");
    } else{
        char text[] = "esta es mi primera escritura por codigo en c";
        fprintf(fd,"Texto escrito: %s", text);
    }

    fclose(fd);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

ARCHIVOS

```c
#include <stdio.h>

int main(){

    FILE *fd;

    char direccion[] = "file.txt";

    fd = fopen(direccion,"a");

    if (fd == NULL){
        printf("archivo no existe");
    } else{
        char letras;
        while ((letras = getchar()) != '\n'){
                fputc(letras,fd);
        }
    }

    fclose(fd);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

FEOF

```c
#include <stdio.h>

int main(){

    char nombre[20];
    printf("digite el nombre: ");
    do
    {
        scanf("%s",nombre);
        if (!feof(stdin)){
                printf("\n%s",nombre);
        }
    } while (!feof(stdin));

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

HEADERS

```c
#include <stdio.h>
#include "suma.h"

int main(){

    int num1 = 3, num2 = 5;

    int result = suma(&num1,&num2);

    printf("el resultado de la suma
    es: %i",result);

    return 0;
}
```

```c
#include <stdio.h>

int suma(int *p1, int *p2){
    int sum;
    sum = *p1 + *p2;

    return sum;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

MEMORIA ESTÁTICA Y DINÁMICA

1	**1**	**1**	**1**						
							1	1	1
			1	1	1	1	h	h	h
h	h	h	h	h	h	h	1	1	1
h	h	h	h	h	h	h	h		

UNIVERSIDAD EIA®

Ser, Saber y Servir

MEMORIA ESTÁTICA Y DINÁMICA MALLOC

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

int main(){

    int* i;
    i = (int*)malloc(sizeof(int)*8);
    if(i != NULL){
        printf("la memoria se asigno correctamente");
    }else{
        printf("la memoria no se asigno correctamente");
    }

    printf("\neste es el puntero %p: ",i);

    free(i);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

MEMORIA ESTÁTICA Y DINÁMICA MALLOC

ESTÁTICA

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

int main(){

    int num = 20;
    printf("%i", num);

    return 0;
}
```

DINÁMICA

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

int main(){

    int *a;
    a = malloc(sizeof(int));

    *a = 5;

    printf("%i",*a);

    free(a);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

MEMORIA ESTÁTICA Y DINÁMICA
REALLOC

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

int main(){

    int *v1 , *v2;
    v1 = malloc(3*sizeof(int));

    v1[0] = 23;
    v1[1] = 15;
    v1[2] = 75;

    for (int i= 0; i < 3; i++){
        printf("\n%i",v1[i]);
    }

    v2 = realloc(v1,5*sizeof(int));

    v2[3] = 55;
    v2[4] = 53;

    for (int i= 0; i < 5; i++){
        printf("\n%i",v2[i]);
    }

    free(v2);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir