



UTN – FRRO
Ingeniería en Sistemas de Información

Cátedra: Sintaxis y Semántica de los Lenguajes

GUÍA PRÁCTICA LENGUAJE C

Plan 2023

Descripción breve

Contiene una serie de ejercicios, agrupados por temas, para poner en práctica la Sintaxis y Semántica del lenguaje C.

Contenido

CONSTANTES, VARIABLES, OPERADORES Y EXPRESIONES	2
FUNCIONES DE ENTRADA Y SALIDA	7
SENTENCIAS DE CONTROL DE FLUJO: if, switch, for, while, do while	9
ARREGLOS Y CADENAS	13
ESTRUCTURAS Y ARREGLOS DE ESTRUCTURAS	17
FUNCIONES (PASAJE DE PARÁMETROS POR VALOR)	19
PUNTEROS	20
FUNCIONES (PASAJE DE PARÁMETROS POR REFERENCIA - PUNTEROS)	22

CONSTANTES, VARIABLES, OPERADORES Y EXPRESIONES

- 1- Definir la constante simbólica DOLAR dándole el valor actual de la moneda.
- 2- ¿Cuál o cuáles de las siguientes sentencias son válidas en lenguaje C teniendo en cuenta que VALOR es una constante simbólica numérica entera?
 - a. VALOR + = i;
 - b. printf ("%s", VALOR);
 - c. ++ VALOR;
 - d. printf ("%d", VALOR);
 - e. Ninguna de las anteriores
- 3- Habiendo definido previamente la constante simbólica DOLAR, la misma ¿puede cambiar su valor en tiempo de ejecución? (marque la respuesta correcta)
SI - NO

- 4- Decir si las siguientes expresiones son válidas:

- a. #define PI 3.141592, NUMERO_E 2.718281
- b. #define FIESTA "7 de julio \"San Fermín\""

- 5- Dada la siguiente directiva al preprocesador:

```
#define CATEDRA "Sintaxis y Semántica de los Lenguajes"
```

Si escribimos:

```
printf ("Bienvenidos a %s !!", CATEDRA);
```

por pantalla se verá: R: _____

- 6- Decir cuáles de los siguientes identificadores son válidos en el lenguaje C, y si no lo son decir porque:

a) registro1	d) return	g) nom y dir	j) 123-45-6
b) 1registro	e) \$impuesto	h) nom_y_dir	k) nom*ap
c) archivo_3	f) nombre	i) nom-y-dir	l) num1_art

- 7- Determinar cuáles de los siguientes valores numéricos son constantes válidas. Si es una constante válida especificar si es entera o real.

a) 0.5	c) 9.3e12	e) 12345678
--------	-----------	-------------

b) 27,822	d) 9.3e-12	f) 127.09
-----------	------------	-----------

8- Determinar cuáles de las siguientes son constantes de carácter válidas.

a) 'a'	c) '\n'	e) '\\'	g) 'T'	i) 'xyz'
b) 'S'	d) '/n'	f) '\a'	h) '\0'	j) "a"

9- Determinar cuáles de las siguientes son constantes de cadena de caracteres válidas.

- a. '8:15 P.M.'
- b. "rojo, blanco y negro"
- c. "Nombre:
- d. "Capítulo 3\n \"SUMAS ALGEBRAICAS\""
- e. "2000 ROSARIO"
- f. "El profesor dijo: "Por favor, presten atención""

10- Escribir las declaraciones apropiadas para cada grupo de variables:

Variables enteras: p, q

R: _____

Variables de coma flotante: x, y, z

R: _____

Variables de carácter: a, b, c

R: _____

Variable entera larga: cont

R: _____

11- Escribir las declaraciones apropiadas y asignar los valores iniciales dados para cada grupo de variables.

Variables de coma flotante: a = -8.2, b = 0.005

R: _____

Variables enteras: x = 129, y = 87, z = -22

R: _____

Variables de carácter: c1 = 'w', c2 = '*'

R: _____

Variable de carácter: eol = caracter de nueva línea

R: _____

12- Cuál es el resultado de cada una de las siguientes expresiones:

a) $!(3 > 3 \ \&\& \ 5 != 4 \ || \ 9 = 9)$ R: _____

b) $!(7 > 8) \ \&\& \ 7 > 8 \ || \ 7 > 8$ R: _____

c) $13/13*2*5+5\%2$ R: _____

13- Construir una expresión para cada una de las siguientes condiciones:

a) El valor de la variable **num** no está entre 1 y 5

R: _____

b) El valor de la variable **car** no es el carácter 'A' ni el carácter 'Z'.

R: _____

c) La variable **num** está entre 0 y 9 pero no es 3.

R: _____

14- Dadas las siguientes declaraciones y asignaciones: **int a = 2, b = 3, c = 4**; evaluar las siguientes expresiones y determinar que quedará almacenado en **result** en cada caso.

- a) `result = a * b % c;`
- b) `result = 3 * (a -= 1) + 1 % a;`
- c) `result = (a > b) || (b == c);`

15- ¿Qué valor contendrán las variables **s** y **c** luego de ejecutarse las secuencias de asignaciones que se mencionan en cada caso?

- | | | |
|--------------------------------|----------------------------|------------------------|
| a) <code>c = 0;</code> | b) <code>c = 0;</code> | c) <code>c = 4;</code> |
| <code>s = (++ c) + 1 ;</code> | <code>s = c ++ + 1;</code> | <code>c* = 2 ;</code> |

R: _____ R: _____ R: _____

16- ¿Qué valor queda en cada variable involucrada, después de ejecutar el enunciado que se indica, suponiendo que antes de ello todas las variables contienen el valor 5?

- a) `product * = x++ ;`

R: _____

- b) `result = ++x + x;`

R: _____

17- Escribir al menos de tres formas distintas la operación de incrementar en 1 la variable **cont**.

18- Completar las líneas de punto:

- a) El operador..... permite realizar una operación si una condición es verdadera y otra si es falsa.
- b) El operador..... devuelve el resto de la división entera.
- c) Una expresión lógica es verdadera si
- d) Toda sentencia termina en.....
- e) Un programa escrito en lenguaje C está íntegramente formado por
- f) Todo programa en C comienza su ejecución en la función

19- Responder a cada una de las siguientes afirmaciones con **V (verdadero)** o **F (falso)** según corresponda. Justificar.

a. El operador % puede ser utilizado con cualquier tipo de operandos.

R: _____

b. En la expresión **a= ('4' > 4)** la variable a asume el valor 1.

R: _____

c. La expresión **(x > y || a < b)** es verdadera ya sea que (x > y) es verdadero o (a < b) es verdadero.

R: _____

d. Una variable local puede ser definida en cualquier lugar dentro de una función.

R: _____

20- Escribir una instrucción donde dada una variable, se imprima la letra "n" si el número contenido en la variable es < 0 y "Z" si el número es >= 0.

R: _____

21- Asignarle a la variable **var** el contenido de la variable **res** y luego disminuir **res** en 1(unos).

R: _____

22- Escribir en una sola instrucción donde se pueda asignar a la variable **p** un 1 si **p** es impar o un 0 si **p** es par, esto en el caso de que **p** sea mayor que 20, sino asignarle un 6.

FUNCIONES DE ENTRADA Y SALIDA

23- Escribir los enunciados necesarios para ingresar por teclado las variables:

- x de tipo entera e y de tipo flotante, separadas por una , (**coma**)
- x de tipo caracter e y de tipo entera, separadas por una / (**barra**)
- x de tipo entero largo e y de tipo caracter, separadas por un **blanco**

24- Escribir los enunciados necesarios para mostrar:

a) La mesa número **num** está servida.

R: _____

b) Hoy es el día nro **num** de la semana.

R: _____

c) En la columna nro **num** hay un error.

R: _____

d) Los números son: **n1, n2, n3**.

R: _____

e) El importe de la factura es **num**.

R: _____

25- Escribir un programa que ingrese un carácter y muestre en pantalla el número decimal que le corresponde en la tabla **ASCII**.

26- Decir si **scanf ()**, **printf ()**, **getchar ()** son:

a) palabras reservadas del lenguaje C. R: SI / NO

b) funciones cuyos prototipos se encuentran en algún archivo de cabecera cuya extensión es .h. R: SI / NO

c) funciones creadas por el programador. R: SI / NO

27- Leer una cadena de caracteres y un valor entero ambos separados por un * almacenándolos en las variables **cad** y **num** respectivamente.

- 28- Escribir un programa que ingrese un entero y muestre en forma tabulada el número, su cuadrado y su cubo.
- 29- Escribir un programa que lea el radio de un círculo y que imprima el diámetro del mismo, su circunferencia y su área. Utilice el valor constante 3.14159 para "PI" y defina al mismo como constante simbólica.
- 30- Escriba un programa que borre la pantalla y escriba en la primera línea su nombre y en la segunda su apellido:

SENTENCIAS DE CONTROL DE FLUJO: if, switch, for, while, do while

- 31- Hacer el algoritmo que nos permita introducir un número por teclado y nos informe si es positivo o negativo.
- 32- Hacer el algoritmo que nos permita introducir un número por teclado y nos informe si es par o impar.
- 33- Escribir el algoritmo que me permita leer un número decimal que representa una cantidad de grados Celsius y convierta dicho valor a la cantidad equivalente en grados Fahrenheit. La salida del programa puede ser de la siguiente forma: 100 grados Celsius son 212 grados Fahrenheit.
- 34- Diseñar el algoritmo necesario para que habiéndose leído el valor de 2 variables **NUM1** y **NUM2** se intercambien los valores de las variables, es decir que el valor que tenía **NUM1** ahora lo contenga **NUM2** y viceversa.
- 35- Escribir un enunciado C para conseguir lo siguiente:
- a) Ingresar un texto carácter a carácter hasta que se ingrese '#', sin almacenar dicho texto.
 - b) Ingresar un texto carácter a carácter hasta que se active el **EOF**, sin almacenar dicho texto (para activar el EOF utilizar **ctrl+Z** en entorno windows y **ctrl+D** en compiladores on line).
- 36- Detectar y justificar los errores en cada uno de los siguientes programas:
- a.

```
#include <stdio.h>
main() {
    /* Aquí va el cuerpo del programa */
    Printf( "Hola mundo\n" );
}
```
 - b.

```
#include <stdio.h>
main() {
    int numero;
    numero = 2
    printf( "El valor es %i" Numero );
}
```
 - c.

```
#include <stdio.h>
main() {
    int a;
    a = 5;
    c += a +5;
```

```

        printf ("c = %d \n", c);
    }

```

37- Qué resultado se obtiene de evaluar los siguientes segmentos de programa:

a. `main() {`
 `int a = 3, b = 5, c = 4;`
 `if (a < b > c) printf ("Correcto");`
 `else printf ("Incorrecto");`
 `}`

R: _____

b. `main() {`
 `int a = 10, b= 3;`
 `(a % b == 0)? printf (" Azul\n"): printf (" Rojo\n");`
 `}`

R: _____

c. `main() {`
 `int a = 10, b= 5;`
 `char c;`
 `c = (a % b == 0)? 'A' : 'B';`
 `printf (" %c ", c); }`

R: _____

d. `main() {`
 `char c; int i; float f;`
 `c = i = f = 'A' + 5;`
 `printf (" %c %d %f ", c, i, f); }`

R: _____

e. `main() {`
 `int expresion, x = 1;`
 `expresion = ! (1 == x) && 4)`
 `if (expresion)`
 `printf ("El resultado de la expresión es verdadero");`
 `else`
 `printf ("El resultado de la expresión es falso");`

```

x = 4;
for( ; x < 10; x += 3);
printf ("%d", x / 2);
x = 10;
do {
    printf ("%d", x--);
} while ( x >5);
}

```

R: _____

- 38- Mostrar una tabla que contenga los valores de peso y costo, variando el peso desde 2 hasta 20 con un incremento de 6, si el costo mínimo es de \$50 y se incrementa en \$15 y cada 6 kg. Utilizar un bucle for y constantes simbólicas donde se considere útil.
- 39- Incrementar una variable entera j desde 0 hasta n y luego mostrar el resultado. Hacerlo utilizando 3 bucles diferentes.
- 40- En la expresión **for(i=1; i==10; i+=2);** cuántas veces se evalúa la condición.
- 41- Ingresar **n** números y calcular su media. **media = (nromayor + nromenor) / 2.**
- 42- Escribir el algoritmo necesario para calcular y mostrar el cuadrado de un número ingresado por teclado. El número debe ser mayor que cero, en caso de error que aparezca el mensaje "ERROR, el número debe ser mayor que cero". El programa termina su ejecución cuando se ingresa un **0**.
- 43- De 10 números ingresados indicar cuántos son mayores que cero y cuántos son menores que cero.
- 44- Leer sucesivamente de teclado, hasta que aparezca un número comprendido entre 1 y 5. Desarrollar el algoritmo usando la función:
- a) **getchar()**
b) **scanf()**
- 45- Codificar en C un programa que lea 20 caracteres indique cuántas "**a**" se ingresaron, cuantas "**e, i, o, u**".
- 46- Realizar un algoritmo que permita ingresar un número correspondiente a los días de una semana y muestre el nombre del día. Que se permita trabajar hasta que el usuario indique lo contrario.

- 47- Hacer el algoritmo que muestre los números pares entre el 1 y el 100 a razón de 6 por línea separados por 3 blancos.
- 48- Codificar en C un programa que muestre los múltiplos de 2 y de 3 y de ambos comprendidos entre 1 y 100.
- 49- Hacer un algoritmo que muestre todos los números pares que hay desde la unidad hasta un número que introducimos por teclado. El programa debe poder ejecutarse mientras el usuario lo requiera.
- 50- Codificar en C un programa que muestre los números primos comprendidos entre 0 y 100.
- 51- Codificar en C un algoritmo que permita ingresar 10 números, ninguno de ellos igual a cero. Se pide sumar los positivos, obtener el producto de los negativos y luego mostrar ambos resultados.
- 52- Calcular la media de los números no negativos de una lista de n números ingresados por teclado (n se ingresa por teclado).
- 53- Dados 3 números donde el primero y el último son límites de un intervalo, indicar si el tercero pertenece a dicho intervalo.
- 54- Realizar la tabla de multiplicar de un número entre 0 y 10 de forma que se visualice de la siguiente forma:
4x1= 4
4x2= 8
.....
- 55- Hacer un algoritmo que muestre el mayor y el menor de una serie de 5 números que vamos introduciendo por teclado.
- 56- Escribir un programa que solicite el ingreso de un número entero y a partir de él, genere una cuenta regresiva hasta cero, mostrando los sucesivos valores hasta el último luego de lo cual emitirá un pitido.
- 57- Escribir un programa que genere y muestre en pantalla la tabla **ASCII**.
- 58- Ingresar un texto de caracteres utilizando la función **getchar()**. Indicar la cantidad de caracteres, palabras, y líneas que lo forman. Mostrar lo pedido con carteles aclaratorios. Considerar como separadores de palabras válidos: espacio, tabulador y <enter> y tener en cuenta que contar palabras no es contar cantidad de separadores.

ARREGLOS Y CADENAS

59- Dado el siguiente programa indicar qué resultado se obtiene.

```
#include<stdio.h>
main(){
int vec[8]= {1,2,-2,1,3,-1,5,10}, i, var;
var=1;
for( i = 0; i <= 7; ++i )
    if(vec[i]>0) var = var * vec[i];
printf ( " %d", var);
}
```

R: _____

60- Dada la siguiente declaración e inicialización del arreglo A:

```
int A[5] = { 1,2,3,4,5}
```

Responder:

Valor de A[0]

R: _____

Valor de A[4]

R: _____

61- Dado el siguiente programa indicar qué resultado se obtiene.

```
include<stdio.h>
main() {
int vec[8]= {1,2,-2,1,3,-1,5,10}, i, var;
var= i = 0;
do{ var = var + vec[i];
    ++i;
} while (vec[i] >0);
printf ( " %d", var);
}
```

R: _____

62- Dado el siguiente programa indicar qué resultado se obtiene.

```
#include<stdio.h>
main() {
int vec[11]= {1,2,-2,1,3,-1,5,10,-5,2,3}, i, var;
var= i = 0;
while( vec[i] < 10) {
    var = var + vec[i];
    ++i;
}
printf ( " %d", var);}
}
```

R: _____

- 63- Escribir un programa que permita el ingreso de 1000 números, los almacene en un arreglo, luego ingrese otro número y determine entre los anteriores cuales son mayores o iguales al último número ingresado. Con los mayores calcular su sumatoria y mostrar:

LA SUMATORIA DE LOS NÚMEROS MAYORES QUE *nro_ingresado* ES *sumatoria*.

- 64- Escribir un programa que ingrese y almacene 1000 números y que los muestre ordenados según un indicador de criterio de ordenamiento que también debe ingresarse por teclado.

- 65- Escribir un programa que cargue un arreglo bidimensional numérico e indique la fila y la columna en la que se encuentra almacenado el número de mayor valor absoluto.

- 66- Escribir un programa que ingrese 10 números enteros, los cargue en un arreglo unidimensional, busque el mayor de ellos, calcule el promedio y muestre:

LOS NÚMEROS INGRESADOS SON :

EL MAYOR DE ELLOS ES:

SU PROMEDIO ES :

- 67- Escribir un programa que cree un arreglo de 26 elementos y que además genere y guarde en ellas 26 las letras minúsculas del alfabeto.

- 68- Escribir un programa que ingrese una cadena, cuente cuántos de sus elementos son dígitos, cuántos de ellos son blancos y muestre:

LA CANTIDAD DE DÍGITOS QUE APARECEN EN **cadena** ES **cant_digitos** Y LA CANTIDAD DE BLANCOS ES **cant_blanco**

69- Sumar números en un vector de 10 elementos enteros hasta que el usuario lo desee.

Cuando finalicen las sumas visualizar por pantalla lo siguiente:

Elemento 0 cantidad total

.....

Elemento 9 cantidad total

70- Leer dos vectores A y B de dimensión 10. Generar un tercer vector C de 10 elementos donde la componente C[i] sea igual al mínimo valor de A[i] y B[j].

Mostrar los tres vectores.

71- Leer 20 números enteros positivos. Mostrar el vector tal como fue ingresado y luego mostrar el vector ordenado en forma decreciente.

72- Leer 20 números enteros (positivos y negativos) distintos de cero. Mostrar el vector tal como fue ingresado y luego mostrar los positivos ordenados en forma decreciente y por último mostrar los negativos ordenados en forma creciente.

73- Se dispone de dos cadenas de caracteres cad1 y cad2 y se ejecutan las siguientes funciones:

- **strcpy** (cad1, "ANTONIO");
- **strcpy** (cad2, "ANTENA");

Para la expresión strcmp (cad1, cad2); ¿Cuál de las siguientes afirmaciones es cierta?

- a) Lo que devuelve la expresión dependerá de la dimensión de cad1 y cad2.
- b) La expresión devuelve un nº menor que cero.
- c) La expresión devuelve un nº mayor que cero.
- d) Lo que devuelve la expresión dependerá de la dimensión de cad1.
- e) Lo que devuelve la expresión dependerá de la dimensión de cad2.

74- Decir cuál será la salida de los siguientes programas:

```
a)
main() {
char cad1[ ]= "Examen";
char cad2[ ]= "Parcial";
printf ("%s\n", strrev (cad2));
printf ("%d", strlen (cad1)); }
```

R: _____

R: _____


```

b)
main() {
char cadena[20];
strcpy (cadena, "5, 89, 123,2");
strrev (cadena);
printf (" %s \n", cadena);
printf (" %c \n", *cadena); }

```

R: _____

R: _____

- 75- Hacer un programa que ingrese dos cadenas y:
 Determine cuál de ellas es mayor en orden alfabético.
 Concatene ambas cadenas.
 Determine cuál es la cadena más larga.
 Invierta la primera cadena ingresada.

El programa deberá mostrar todas las salidas con carteles aclaratorios

LA MAYOR ES

LAS CADENAS CONCATENADAS QUEDAN.....

LA CADENA MÁS LARGA ES.....

LA CADENA cadena INVERTIDA SE LEE

- 76- Escribir un programa que encuentre el número de veces que una palabra dada ocurre en un texto:

Una salida podría ser:

La palabra es "el"

El texto es "el perro, el gato y el canario"

La palabra ocurrió 3 veces.

- 77- Escribir un programa en **C** que solicite el ingreso de un texto, luego ingrese una letra y muestre, ordenadas alfabéticamente, todas las palabras del texto que comienzan con dicha letra. El ingreso de la letra se podrá repetir tantas veces como se desee. No utilizar variables globales. (Cant. Max. Caracteres: 1500 – Cant. Max. Palabras: 100 – Long. Max. Palabra: 15 caracteres)

- 78- Escriba un programa que inserte un carácter en una determinada posición de una cadena.

ESTRUCTURAS Y ARREGLOS DE ESTRUCTURAS

79- Declarar un tipo de estructura de nombre **alumnos** para almacenar la siguiente información:

- **nombre del alumno** **char[20]**
- **nº de legajo** **char[5]**
- **domicilio** **char[15]**
- **turno** **char**

Declarar una variable del tipo alumnos dentro del main y mostrar todos sus campos.

80- Si declaramos fuera del main el tipo de estructura: **struct alumnos{int cod; char nom [10] } alum;** . La variable **alum** tendrá un alcance.....en el programa.

81- Declarar un tipo de estructura de nombre **alumnos** para almacenar la siguiente información:

- **nombre del alumno** **char[20]**
- **turno** **char.**

Dentro del main declarar dos variables del tipo alumnos y asignarles valores a sus respectivos campos. Determinar si ambos alumnos pertenecen al mismo turno o no.

82- Declarar un tipo de estructura de nombre alumnos para almacenar la siguiente información:

- **nombre del alumno** **char[20]**
- **nº de legajo** **char[5]**
- **domicilio** **char[15]**
- **turno** **char**

Luego en el main declarar un arreglo para cada uno de los 300 alumnos de un establecimiento.

Asignar en la 1ª estructura del arreglo los siguientes datos: "Juan Gomez", "123245", "San Juan 1056", 'M'. Finalmente mostrar los datos cargados en la posición 0 del arreglo.

83- De los alumnos de una escuela se registra su apellido, nombre y su altura. **Como máximos hay 400 alumnos.**

Diseñar un algoritmo que indique el nombre del alumno más alto y su altura (sólo uno es el más alto).

84- Se dispone de registros con los siguientes. datos: código (5 caracteres), nombre (20), domicilio (20), Saldo.

Se solicita cargar dichos datos (500 registros como máximo) y luego satisfacer un esquema de consulta interactiva donde al ingresar un código se informan los datos restantes correspondientes.

- 85- Se ingresa el apellido, la nota y el legajo de los 8 mejores alumnos de un curso. Realizar un programa que permita modificar los datos ingresados hasta que el usuario lo determine. Para ello se ingresa el legajo y la nota a modificar. Si no se encuentra el legajo se deberá mostrar un mensaje de error. Finalmente mostrar los datos ordenados por legajo.
- 86- Un profesor de matemática de un establecimiento educativo registra de cada alumno N° de legajo, nombre y promedio. Según el promedio desea saber cuántos alumnos aprobaron (promedio mayor o igual a 7), cuantos rinden en diciembre (promedio menor a 7 y mayor o igual a 4) y cuantos rinden examen en marzo (promedio menor a 4). Además, desea conocer el N° de legajo y nombre del alumno con mejor promedio. El profesor tiene 300 alumnos en sus clases.

FUNCIONES (PASAJE DE PARÁMETROS POR VALOR)

87- Dentro de una función, si dos variables, una local y una global, tienen el mismo nombre, ¿qué afirmación es cierta?

- a. La global prevalece sobre la local.
- b. La local prevalece sobre la global.
- c. Se producirá un error.

88- Las variables definidas fuera de toda función se denominan

89- Diseñar un programa en C para calcular el cubo de los números del 1 al 5 utilizando una función definida por el usuario para realizar el cálculo del cubo. El prototipo de la función es el siguiente:

int cubo (int base);

90- Ingresar dos números enteros y luego presentar el siguiente menú de opciones:

- 1- SUMAR
- 2- RESTAR
- 3- MULTIPLICAR
- 4- DIVIDIR

A esto el usuario debe responder con la opción correspondiente a la operación que desee hacer entre los números y el programa debe devolver el resultado. Utilizar funciones para las llamadas a cada punto del menú de opciones. El programa debe ser iterativo.

91- Escribir un programa que lea el radio de un círculo y que imprima el diámetro del mismo, su circunferencia y su área. Utilice el valor constante 3.14159 para "PI" y defina al mismo como constante simbólica. Utilizar funciones para cada uno de los diferentes cálculos a realizar.

92- Escribir un programa que ingrese 10 números enteros, los cargue en un arreglo unidimensional, y a través de dos funciones busque el mayor de ellos y calcule el promedio y muestre:

LOS NÚMEROS INGRESADOS SON :

EL MAYOR DE ELLOS ES:

SU PROMEDIO ES :

Las funciones a utilizar deben llamarse máximo y media respectivamente. Diseñar los prototipos correspondientes.

PUNTEROS

93- Decir cuál será la salida:

```
#include<stdio.h>
main() {
    int arr[10] = {23, 5, 98, 65, 3, 55, 73, 9, 21, 85}, *p;
    p = arr;
    printf ( " %d\n", arr[(p + 7)]); //R: _____
    printf ( " %d\n", *arr + 3);    //R: _____
    printf ( " %d\n", *p++);        //R: _____
    printf (" %d\n", *(arr + 1));   //R: _____
    printf (" %d\n", (*p)++);       //R: _____
    printf (" %d\n", *p);           //R: _____
    printf (" %d\n", *p++);         //R: _____
    printf (" %d\n", *p); }        //R: _____
```

94- Dado el siguiente fragmento de programa:

```
char u, v = 'A';
char *pu, *pv = &v;
.....
*pv = v + 1;
u = *pv + 1;
pu = &u;
```

y suponiendo que el valor asignado a u se almacena en la dirección F8C y el valor asignado a v se almacena en la dirección F8D, decir:

- a) ¿Qué valor es asignado a **pv**?
- b) ¿Qué valor es representado por ***pv**?
- c) ¿Qué valor es asignado a **u**?
- d) ¿Qué valor es asignado a **pu**?
- e) ¿Qué valor es representado por ***pu**?

95- Sabiendo que en código ASCII la letra 'e' es el valor decimal 101, escribir que muestra cada línea de printf.

```
#include <stdio.h>
int main () {
    char c;
    c='e';
    printf ("%c",c);    //R: _____
    c++;
    printf ("%d",c);    //R: _____
    printf ("%c",c);    //R: _____
}
```

96- ¿Qué valor muestra el último printf?

```
#include <stdio.h>
main() {
int num , * punt , tol;
num=8;
punt= &num;
tol=*punt;
printf ("el total es: %d",tol); //R: _____
}
```

97- Suponiendo que x es un puntero de tipo entero y num es una variable entera, si **x = &num** y **a = *x**, ¿cuánto vale **a**? (marque con una cruz la respuesta correcta).

- a) **a** vale igual a num _____
- b) **a** contiene la dirección de memoria de num _____
- c) **a** contiene la dirección de memoria de x _____

FUNCIONES (PASAJE DE PARÁMETROS POR REFERENCIA - PUNTEROS)

98- Ingresar desde el main una cadena, y a través de una función mostrar cuántos dígitos tiene la cadena y también su longitud. La función recibe dos parámetros, un puntero a char para recibir la cadena, y un puntero a entero que devuelve al main la cantidad de dígitos que tiene la cadena. La función retorna un entero con la cantidad de caracteres que tiene la cadena (su longitud).

El prototipo de la función es el siguiente: **int verificaDigitos (char *, int *);**

99- Ingresar una cadena y por medio de una función determinar si dicha cadena está formada sólo por caracteres alfanuméricos. La función devolverá el valor "SI" en caso afirmativo, o el valor "NO", en caso contrario.

El prototipo de la función es el siguiente: **char * verif (char *);**

100- Desarrollar un programa en lenguaje C, que realice las siguientes acciones:

- Cargue un texto.
- Determine cuál es el porcentaje de vocales sobre los caracteres del texto.
- Determine cuál es el porcentaje de consonantes sobre los caracteres del texto.
- Muestre:

**El texto..... tiene.....% de vocales y
.....% de consonantes.**

Para resolver esta problemática deberá utilizar las siguientes funciones:

- Una función que cargue un texto(que incluya cualquier tipo de caracteres, incluso la nueva línea) cuyo prototipo será:
void carga (char*, int*)
- Una función que solamente determine si un carácter es una vocal (no cuente cantidad de vocales), su prototipo será:
void esVocal (char, int*)
- Una función que solamente determine si un carácter es una consonante (no cuente cantidad de consonantes), con prototipo: **int esConsonante (char)**

101- Desarrollar un programa en lenguaje C, que realice las siguientes acciones:

- Cargue un texto.
- Cambie los dígitos por * (asteriscos).
- Muestre:

**El texto..... cambiado se lee:
.....**

Para resolver esta problemática deberá utilizar las siguientes funciones:

- Una función que cargue un texto(que incluya cualquier tipo de caracteres, incluso la nueva línea) cuyo prototipo será:
char * carga (int*)
- Una función que cambie el texto, su prototipo será:
char* cambia_texto (char *, int)