

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 1

TEMA: Algoritmos. Procedimientos y funciones



Para la resolución de este primer práctico vamos a retomar el diseño de algoritmos visto en la materia “Elementos de Computación y Lógica” usando pseudocódigo.

Realice un análisis minucioso de cada problema planteado y determine:

- ¿Cuáles son los datos de entrada del problema en cuestión?
- Respecto a la salida del algoritmo ¿Cuáles son los resultados esperados?
- En general ¿De qué tipo son estos datos?.
- En el algoritmo, ¿necesita de variables auxiliares? Recuerde que se trata de datos que no representan ni *Entradas* ni *Salidas* sino, son variables extras de apoyo para realizar ciertos cálculos y que como su nombre lo indica, su valor por lo general cambia a medida que se cumplen las instrucciones de un algoritmo.
- Eventualmente (si lo considera conveniente o necesario) también puede hacer uso de constantes. Estas tienen valores que se mantienen fijos durante la vida del algoritmo, por ejemplo un porcentaje de descuento a un producto. La ventaja de su uso, es que si el porcentaje cambia en algún momento, solo se actualiza la constante.
- Si trabaja con funciones o procedimientos, ¿cuáles son los parámetros o argumentos que necesita para resolver la tarea específica que le toca?. No olvide el tipo. En el caso exclusivo de funciones, ¿cual es el tipo de dato de la variable que retorna?



Sugerencia: Después de escribir la solución algorítmica de un problema, analice qué cambios haría para tener una mejor solución del mismo. El contraste entre ambas formas resulta instructivo para encontrar la solución más adecuada.

Nadie aprende a dibujar mirando a un profesor que dibuja muy bien... nadie aprende a escribir ni a pensar escuchando a un hombre que habla y piensa bien. Hay que intentarlo, hacer, rehacer, hasta que el oficio se "haga carne".

(Alain, 1932; citado en Meirieu, 2016).

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

1. Las Manos del Mundo

La 20° edición de la popular feria de artesanías “Las Manos del Mundo” regresó a la provincia de Tucumán luego del confinamiento y abrió sus puertas desde el 9 hasta el 15 de agosto. La feria albergó artesanos de diferentes provincias del país que llevan su arte por diferentes lugares del mundo.

Cada artesano paga un monto en concepto de uso/alquiler del espacio donde exhibe sus productos. El mismo se calcula a partir de la categoría a la que pertenecen y de la dimensión del box que ocupa.

Las tarifas por día son las siguientes:

- Telas y tejidos. \$850
- Vidrio y metales. \$750
- Cuero. \$980
- Productos regionales. \$1100

Adicionalmente, por el tamaño del box que ocupa cada artesano, se cobra un porcentaje sobre el monto a pagar por la totalidad de días que dura la feria:

- Box chico (2 mts x 1,5 mts). Adicional del 15%.
- Box mediano (3 mts x 1,5 mt). Adicional del 20%.
- Box grande(4 mts x 2,5 mts). Adicional del 30%

Diseñe una función que a partir de los datos correspondientes calcule y muestre el monto final que paga cada ocupante por el alquiler de un box. Luego, escriba un algoritmo que calcule el monto total recaudado por el comité organizador durante los días que permaneció abierta la feria para un total de 158 artesanos.

2. Data Entry

Una empresa dedicada al data entry necesita agilizar su trabajo, por este motivo le pidieron al dpto. de sistemas desarrollar una pequeña aplicación que complete los números de celular disponibles en su base de datos. Los mismos deben formarse por 3 partes:

- Característica: La cual puede ser Tucumán (0381), Cordoba (0351) o Buenos Aires (011)
- Prefijo: 15
- Número

Diseñe un algoritmo que a partir de una ciudad (1: Tuc, 2: Cba, 3: Bs As), que se lee por única vez para toda la secuencia original, complete y muestre una secuencia de números de celular completos con la característica correspondiente y el prefijo. Tenga en cuenta que cada número de celular está separado por una “,” y que la secuencia termina en “.”

Ejemplo de la secuencia de números original: 5444111,3100200,4123100.

Ejemplo de secuencia de salida: 0381-155444111,0381-153100200,0381-154123100

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

3. Mínimo Común Múltiplo

El mínimo común múltiplo (mcm) de dos números A y B es *el número más pequeño que es múltiplo de A y múltiplo de B*.

Con el máximo común divisor (MCD) de una pareja de números podemos obtener fácilmente el mínimo común múltiplo de dicha pareja. **El mcm de dos números es igual al producto de los números dividido entre su MCD.**

Por ejemplo, el máximo común divisor de 24 y 36 es 12, por lo tanto el mínimo común múltiplo de 24 y 36 es $(24 \times 36) / 12 = 72$.

Ahora bien, ¿cómo calculamos el MCD de dos números?

El máximo común divisor (MCD) de dos enteros A y B es *el entero más grande que divide tanto a A como a B*. El algoritmo de Euclides es una técnica para encontrar rápidamente el MCD de dos enteros y trabaja de la siguiente forma:

Se divide el número mayor entre el menor.

- Si la división es exacta, el divisor es el MCD.
- Si la división no es exacta, se divide el divisor en el resto obtenido y se continúa de esta forma hasta obtener una división exacta, siendo el último divisor el MCD.

Diseñar una **función que calcule el mcm de dos números enteros**. Realice un algoritmo de prueba. Los números A y B pueden estar desordenados y no pueden ser cero.

4. Diseñando una Calculadora

Se cuenta con una calculadora que tiene un set de instrucciones reducido. Se necesita la ayuda de usted como programador para que diseñe un algoritmo tal que se ingresen dos números y un código de operación para que devuelva el resultado. Para esto debe diseñar y hacer uso de las siguientes funciones:

- suma(a,b), que dados dos números a y b, retorne su suma.
- producto(a,b), que dados dos números a y b, retorne su producto (multiplicación)
- paridadSuma(a,b), que retorna 0 si la suma de a y b es par, y 1 si la suma de a y b es impar.
- selector(op, a, b), donde op será 0, 1, 2. Dependiendo del valor de op, se determina cuál de las 3 operaciones anteriores se debe realizar sobre a y b.

Si op = 0, se debe retornar la suma. Si op = 1, se debe retornar el producto. Y si op = 2, se debe retornar el resultado de la función paridadSuma.

ATENCIÓN: Para obtener el resultado de la función producto utilizar suma (+). Use convenientemente las funciones diseñadas de manera de no repetir cálculos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 2

TEMA: Compilador GCC. Etapas de compilación. Entorno de Programación y Pruebas.
Concepto de Tipos de Datos. Tipos de Datos Simples. Constantes y variables. Entrada y Salida básicas.



En este práctico se verá el proceso de compilación completo (en todas sus etapas) para traducir archivos de código fuente escritos en el lenguaje de programación C usando el programa compilador gcc del proyecto GNU. Además, en una segunda parte se trabajarán ejercicios sencillos para conocer el entorno de programación en Visual Studio Code.



Nota: La cátedra usará Visual studio Code por lo cual sugerimos su uso para programar en C pero pueden optar por la opción que les sea de mayor agrado.

OPERADORES BINARIOS:	
Signo	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo

OPERADORES UNARIOS	
Signo	Operación
++	Incremento (suma 1)
--	Decremento (resta 1)
-	Cambio de signo

Operadores de asignación	
Signo	Operación
=	Asignación simple
+=	Suma
-=	Resta
*=	Multiplicación
/=	División
%=	Módulo

Introducción

C es un lenguaje compilado, entonces de forma básica lo que se debe hacer para escribir y ejecutar un programa en C es:

- Utilizar un editor de texto para escribir el programa.
- Enviar el programa al compilador (GCC) el cual se encarga de generar el código ejecutable.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

PRIMERA PARTE: Pruebas en consola + bloc de notas

1) Compilando nuestro código en lenguaje C.

- a) Abra la terminal de comandos de su SO. En Windows lo encontrará como CMD. Luego para verificar que GCC (el compilador de C) se encuentre correctamente instalado, ejecute el siguiente comando: **gcc --version**
En la consola deberemos ver la versión de GCC instalada.

- b) En un bloc de notas escriba el siguiente código escrito en lenguaje C y luego guarde con el nombre y extensión: hola.c

```
1      #include <stdio.h>
2      int main(void)
3      {
4          printf ("Hola mundo!\n");
5          return 0;
6      }
```

- c) En la terminal ejecute el siguiente comando para compilar el código escrito en C y generar un ejecutable llamado hola.exe: **gcc -Wall hola.c -o hola**
Controle que el ejecutable se haya generado.
- d) En el bloc de notas borre el “;” de la línea 4 del código y luego guarde. Ejecute nuevamente el comando para compilar.
¿El compilador le marca algún error? ¿Le indica en qué línea se encuentra el mismo? ¿Se genera el archivo ejecutable? Agregue nuevamente el “;”
- e) Agregue antes de la línea 4, la declaración de una variable de tipo entera con el nombre edad y valor 18: **int edad=18;**

```
..
4      int edad=18;
5      printf ("Hola mundo!\n");
6      return 0;
...
```

Guarde y ejecute nuevamente el comando para compilar.

¿El compilador le indica algún error o advertencia? ¿Por qué? ¿Qué comando permite ver esto?
¿Se genera el ejecutable?

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

2) Compilación etapa por etapa.

a) Abra un bloc de notas y escriba el siguiente código. Guarde con el nombre **monto.c**

```
#include <stdio.h>
#define IVA 0.21

int main(void)
{
    float monto, precio; precio=325.50; //declaro variables.
    monto= precio + (IVA * precio);      //calculo monto a pagar
    printf ("Monto a pagar= %f\n",monto); //muestro en pantalla
    return 0;
}
```

b) Etapa de Preprocesamiento: “.i”

Ejecute en la terminal el siguiente comando para realizar la etapa de preprocesamiento:

gcc -E monto.c -o monto.i

¿Se generó un nuevo archivo? ¿Puede visualizar el código en el nuevo archivo generado? ¿Se encuentran los comentarios en el nuevo archivo? ¿Qué pasó con la constante IVA?

c) Etapa de Compilación: “.s”

Ejecute en la terminal el siguiente comando para realizar la etapa de compilación partiendo del archivo preprocesado: **gcc -S monto.i -o monto.s**

¿Se generó un nuevo archivo? ¿Puede visualizar parte de su código en el nuevo archivo generado?

d) Etapa de Ensamblado: “.o”

Ejecute en la terminal el siguiente comando para realizar la etapa de ensamblado partiendo del archivo previamente compilado: **gcc -c monto.s -o monto.o**

¿Se generó un nuevo archivo? ¿Puede interpretar el contenido del nuevo archivo?

e) Etapa de Enlazado:

Para generar el ejecutable partiendo del archivo previamente ensamblado ejecute en la terminal el comando: **gcc monto.o -o monto**

¿Se generó un nuevo archivo? ¿Qué extensión tiene?

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

SEGUNDA PARTE: Pruebas en entorno de programación.

1. Incrementando y decrementando

1. Declare una variable **incremento** de tipo entera.
2. Declare una variable **decremento** de tipo entera.
3. Solicite por pantalla al usuario que ingrese dos enteros y almacene los mismos en **incremento y decremento**.
4. Incremente en uno la variable incremento (hágalo de dos formas distintas).
5. Decremente en uno la variable decremento (hágalo de dos formas distintas).
6. Muestre por pantalla el contenido de las variables **incremento y decremento**.

2. Operando

1. Declare dos variables **num1** y **num2** de tipo entero.
2. Pida por pantalla al usuario que ingrese un entero y almacene el mismo en **num1**. Pida por pantalla al usuario que ingrese un entero y almacene el mismo en **num2**.
3. Muestre por pantalla los números ingresados por el usuario.
4. Declare una variable **suma** de tipo entero.
5. Asigne a suma el resultado de la suma entre num1 y num2.
6. Muestre por pantalla el contenido de la variable suma.
7. En num1 ponga el contenido de num2 y en num2 lo que tenía num1, muestre en pantalla las variables num1 y num2.
8. Muestre el resto de la división entre num2 y num1.

3. Descuentos

Hernán tiene el negocio de venta de bebidas “El Escollo” y realiza diferentes descuentos en sus productos. Para automatizar los cálculos, se necesita que escriba un programa que pida al usuario ingresar el precio del producto y el porcentaje de descuento que desea realizar. A esto debe aplicarle el IVA del 21%.

Muestre por pantalla el precio final calculado con el descuento con y sin IVA (muestre solo con 2 decimales). Declare IVA como una constante.

4. Promedio

Una maestra de primer grado necesita sacar los promedios de sus alumnos, los cuales tienen 3 materias: Lengua, Matemáticas y Ciencias. Para ayudarla, escriba un programa que pida al usuario ingresar las calificaciones correspondientes a las tres materias, calcule y muestre el promedio obtenido. Aclaración: la maestra califica sobre 10, usando 2 decimales.

5. Volumen

Eugenia necesita sacar el volumen de un cilindro y cuenta con las siguientes medidas: altura (h) en metros y el radio (r) en milímetros. Escriba un programa que resuelva su problema, calculando el volumen en centímetros.

Dato: El volumen de un cilindro es $\pi r^2 h$.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 3

TEMA: Estructuras de Selección. Implementación en lenguaje C

Por defecto, las instrucciones de un programa se ejecutan secuencialmente, sin embargo, al describir la resolución de un problema, es normal que tengamos que tener en cuenta condiciones que influyen sobre la secuencia de pasos que hay que dar para resolver el problema. Esto es, según se cumplan o no determinadas condiciones, la secuencia de pasos involucrada en la realización de una tarea será diferente.

Las estructuras de control condicionales o selectivas nos permiten decidir qué ejecutar y qué no en un programa. Ejemplo típico: Realizar una división sólo si el divisor es distinto de cero.

C proporciona tres tipos de estructuras de selección:

if ejecuta una o varias acciones si la condición es verdadera y omite acciones para cuando la condición es falsa.

if/else ejecuta una acción si la condición es verdadera o ejecuta una acción diferente, si la condición es falsa.

switch ejecuta una entre muchas acciones diferentes, dependiendo del valor de una expresión.



Para el desarrollo del trabajo práctico, en todos los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.



Es importante señalar que, para poder desarrollar capacidades que permitan encarar la resolución de un problema de manera eficaz e independiente del lenguaje de programación es recomendable realizar primero el algoritmo, ya que es aquí donde se encuentra la solución de la problemática en cuestión. Luego, la etapa de codificar dicho algoritmo consistirá en adaptar cada uno de sus elementos al lenguaje de programación en el que se desee implementar.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

1. Triángulos

A partir de los lados a, b y c de un triángulo en centímetros, indique si el mismo es equilátero (tres lados iguales), isósceles (solo dos lados iguales) o escaleno (tres lados distintos). Además calcule y muestre su perímetro.

2. Monotributo

Puede ser monotributista quien vende productos (cosas muebles), presta servicios o integra una cooperativa de trabajo. Con una cuota fija mensual se cubre el pago de las obligaciones impositivas y de seguridad social. Para poder inscribirse se deben cumplir con determinados valores de parámetros como energía eléctrica consumida e ingresos brutos anuales.

Energía eléctrica consumida anualmente	Ingresos brutos anuales	Categoría
Hasta 3330 Kw	\$748.382,07	A
Hasta 5000 Kw	\$1.112.459,83	B
Hasta 6700 Kw	\$1.557.443,75	C
Hasta 10000 Kw	\$1.934.273,04	D
Hasta 20000 Kw	\$3.416.526,83	G
Hasta 20000 Kw	\$4.229.985,60	H

Determine a partir **del cumplimiento de ambos parámetros** la categoría del monotributo a la que debe inscribirse una persona. Tener en cuenta que si se supera el consumo anual de 20000 kilowatts o si los ingresos brutos superan los \$4.229.985,60 no corresponde la inscripción como monotributista sino en el Régimen General y debe informarle al usuario que consulta.

3. Inscripciones 2023

El instituto privado los lapachos ha comenzado con la matriculación para el periodo lectivo 2023. El valor de la matrícula se ha fijado de acuerdo al nivel en el que se inscribe un estudiante:

- **Nivel Inicial:** Salas de 3, 4 y 5 años. \$4800
- **Nivel Primario:** 1° grado a 6° grado. \$6250
- **Nivel Secundario:** 1° año a 6° año. \$6800
- **Nivel Superior:**
 - Profesorado de Educación física. \$7100
 - Profesorado de Lengua y Profesorado de Matemática. \$8000
 - Profesorado de Inglés. \$9500

Si se abona en 2 partes tiene un recargo del 10%, si se abona en 3 partes tiene un recargo del 15% y si se abona en 1 pago se hace un descuento del 15%. Desde el área de tesorería desean saber cuál es el valor final de la matrícula que debe pagar un estudiante.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

4. Mercado en tu Barrio

El programa municipal acerca a los vecinos de la ciudad productos de primera necesidad a precios accesibles, a través de diferentes carpas donde pequeños comerciantes ofrecen mercadería y artículos de limpieza en diferentes puntos de la capital.

El mercado se organiza de manera que por cada rubro de productos que compra una persona se calcula el precio y se pega el mismo en una bolsa cerrada. El cliente recorre el mercado y va llenando su carrito con una bolsa por cada rubro que compra. Al finalizar el recorrido se dirige a una caja general donde se calcula el monto total que debe pagar en función de la variedad de productos que compró.

Según la época del mes desde la coordinación del programa se organizan descuentos por rubro de productos que se aplican al momento del cálculo final. Esta semana, los descuentos son:

Rubro	Descuento
Carnes: de novillo, cerdo, pollo y pescado	15%
Pastas y panificados	10%
Verduras y frutas	5%
Productos de forrajería	15%
Lácteos y huevos	10%
Artículos de limpieza	-

Para agilizar la gestión del mercado, desde la coordinación desean contar con un programa que calcule el monto total a pagar por un cliente sabiendo que no siempre realiza compras para todos los rubros.

5. Previaje

Se trata de un programa de preventa turística que te reintegra el 50% del valor de tu viaje, para viajar y disfrutar de todos los destinos de Argentina. El beneficio aprobado se acreditará en una Tarjeta Precargada emitida por el Banco de la Nación Argentina y en la Billetera Electrónica BNA+ para gastarlo en todo el sector turístico argentino desde la fecha de inicio de tu viaje y hasta el 31/12/2022.

¿Cuáles son los requisitos para acceder al beneficio?

Se mayor de 18 años, tener CUIL o CUIT, domicilio en Argentina y cuenta en Mi Argentina con validación de identidad por datos biométricos (nivel 3).

¿Hay límite de reintegro?

El límite de reintegro es de \$100.000 por persona.

Y hay un monto mínimo de reintegro, que es de \$5.000 por persona (por lo cual, el monto mínimo de gasto es de \$10.000)

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

¿Qué tipo de comprobantes se aceptan?

- Factura B, C cuando el rubro sea agencias de viaje, alojamiento, alquiler de automóviles, museos y parques temáticos, entre otros.
- Ticket aéreo y/o pasaje en caso de servicio terrestre nacional para rubro transporte.

Diseñe un algoritmo que controle si una persona cumple con los requisitos, de no contar con alguno informar de cuál se trata. Para el caso que cumpla, a partir del rubro y tipo de comprobante informe si una persona cuenta o no con el beneficio aprobado, de contar con el mismo muestre el crédito a favor.

Algunos datos de prueba

1. Triángulos

a	b	c	Salida
5	5	2	El triangulo es isosceles
3	8	8	El triangulo es isosceles
4	7	4	El triangulo es isosceles
5	5	5	El triangulo es equilatero
7	8	5	El triangulo es escaleno

2. Monotributo

kw	Ingresos	Salida
3330	\$748.385,00	Pertenece a la categoría B
6500	\$1.200.000,00	Pertenece a la categoría C
20000	\$4.350.000,00	Pertenece al Régimen General, supera ingresos
20100	\$3.416.526,83	Pertenece al Régimen General, supera kw
20000	\$4.229.985,60	Pertenece a la categoría H

3. Inscripciones 2023

Nivel	Profesorado	Cuotas	Salida
Inicial	-	2	\$5.280,00
Primario	-	1	\$5.312,50
Secundario	-	3	\$7.820,00
Superior	Lengua	2	\$8.800,00
Superior	Educ. Física	1	\$6.035,00

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

4. Mercado en tu Barrio

Carnes	Pasta y Panificados	Verduras y Frutas	Forrajería	Lacteos y huevos	Limpieza	Salida
\$500,00	\$200,00	\$125,50	-	-	-	\$724,22
-	-	-	-	\$750,00	\$1.200,00	\$1.875,00
\$1.350,00	-	\$400,00	\$250,00	\$120,00	-	\$1.848,00

5. Previaje

Suponiendo que ya cumple con los requisitos se dejan datos para controlar un comprobante.

Rubro	Tipo	Monto	Salida
Transporte	Ticket Aereo	\$65.000,00	Beneficio APROBADO. Credito disponible: \$32.500
Transporte	Pasaje terrestre	\$48.000,00	Beneficio APROBADO. Credito disponible: \$24.000
Alojamiento	Factura C	\$250.500,00	Beneficio APROBADO. Credito disponible: \$100.000
Museos	Factura B	\$8.000,00	Beneficio APROBADO. Credito disponible: \$0
Excursiones	Factura A	\$16.000,00	Beneficio DESAPROBADO. Ticket no válido
Transporte	Factura B	\$70.000,00	Ticket no compatible con rubro transporte

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 4

TEMA: Estructuras de Iteración. Implementación en lenguaje C

En la vida cotidiana existen situaciones que frecuentemente se resuelven realizando una determinada secuencia de pasos que puede repetirse muchas veces, ejemplo:

- El proceso que seguimos para comer, mientras no se termine la comida.
- El proceso de insertar monedas, hasta que se abone la cantidad necesaria.
- Las operaciones que realizamos para llamar por teléfono, mientras no se logre la comunicación.

Como se puede observar estos son algoritmos que se hacen cotidianamente, pero tienen la particularidad de que la ejecución de alguno de sus pasos puede repetirse muchas veces, mientras no se logre la meta deseada. En C, las instrucciones while, do/while, y for, permiten ejecutar iteraciones, bucles o ciclos. En cada caso se ejecuta un bloque de instrucciones mientras la condición que se evalúa tome valor verdadero.



Para el desarrollo del trabajo práctico, en todos los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.



Es importante señalar que, para poder desarrollar capacidades que permitan encarar la resolución de un problema de manera eficaz e independiente del lenguaje de programación es recomendable realizar primero el algoritmo, ya que es aquí donde se encuentra la solución de la problemática en cuestión. Luego, la etapa de codificar dicho algoritmo consistirá en adaptar cada uno de sus elementos al lenguaje de programación en el que se desee implementar.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

1. VOCALES ABIERTAS

Dada una oración terminada en un punto, escriba un algoritmo que cuente la cantidad total de vocales abiertas (a, e, o) y la cantidad por cada una de ellas.

2. PAR DE NÚMEROS

Dados dos números enteros (que pueden estar desordenados), diseñe un algoritmo para calcular cuántos números hay en el rango entre el menor y el mayor (incluidos los extremos) y cuántos de ellos son pares. Además, calcular la suma de los impares.

Nota: Diseñe una propuesta con la estructura de iteración MIENTRAS y otra con HACER.

3. CADENA RECARGADA

Diseñe un algoritmo que lea una oración terminada en un punto y determine:

- La cantidad de espacios en blanco.
- La cantidad de palabras que comienzan con vocal.
- La cantidad de caracteres que son números.

A tener en cuenta: pruebe que su algoritmo funcione para los siguientes casos:

Caso 1: __ LA __ CASA _ ES _ MUY ___ GRANDE.

Caso 2: ___ .

Caso 3: .

Caso 4: LA ___.

Caso 5: _EL _7 _ Y _9 _SON ___IMPARES.

4. PRIMOS

Dada una secuencia de 10 números enteros, determine si cada uno de ellos es número primo y muestre un mensaje. Además, calcule el total de primos de la secuencia y la suma de los mismos.

Números primos entre 1 y 100: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 y 97.

5. SUBASTA AFIP

La Dirección General de Aduana, entidad dependiente de la Administración Federal de Ingresos Públicos (AFIP), llevará adelante el próximo 18 de septiembre una nueva subasta de celulares, consolas de videojuego y otros artículos tecnológicos que fueron secuestrados en los controles fronterizos del país. Las subastas disponibles son 3:

1. Subasta de 1540 celulares marca Samsung modelo A32. Funcionamiento sin controlar. Expte: 1080
2. Subasta de 720 Playstation 4. Funcionamiento sin controlar. Expte: 1081

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

3. Subasta de lote de 670 auriculares marca Sony, 490 cables USB marca Kyocera y 235 mouse marca Kyocera. Funcionamiento sin controlar. Expte: 1082

Las personas que quieran participar de la subasta lo podrán hacer desde cualquier lugar del país, ya que se realizará mediante la página web de subastas del Banco Ciudad. Los interesados tendrán tiempo para inscribirse hasta 48 horas previas y hacer la transferencia de la caución. Por esta razón, se vuelve indispensable acreditar los siguientes requisitos previstos por la entidad bancaria:

- Para participar en la 1° subasta, sólo podrán inscribirse monotributistas categoría D, F y G.
- Para participar en la 2° subasta, sólo podrán inscribirse monotributistas categoría B, C, y D.
- Para participar en la 3° subasta, sólo podrán inscribirse monotributistas categoría A, B, y C.
- Quienes se hayan inscrito en alguna subasta anteriormente y no hayan presentado una oferta, no podrán participar en esta etapa, al igual que quienes registren deudas en el Veraz.

Diseñe un algoritmo que según los datos proporcionados de un número incierto de personas, informe si cada una de ellas podrá inscribirse en la subasta deseada. Además debe calcular la cantidad de inscriptos por subasta y el total de inscriptos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 5

TEMA: Funciones definidas por el usuario. Funciones de biblioteca.



En lenguaje C, la modularización se lleva a cabo mediante el uso de funciones, que luego cooperarán entre sí para realizar las tareas necesarias de manera de obtener el resultado deseado por el usuario final. Su sintaxis se muestra a continuación:

```
<tipo de retorno><nombre de la función>(<lista de parámetros>){  
    <acciones que realiza la función>  
}
```

Una función puede ser llamada desde cualquier parte del programa que la contiene, posterior a su declaración. Se le invoca con su nombre, seguido de una lista opcional de argumentos (o parámetros). Los argumentos van entre paréntesis y, si hubiera más de uno, separados por comas.

Cuando la función no devuelve valor alguno, se reemplaza el <tipo de retorno> por la palabra reservada void.



Para el desarrollo del trabajo práctico, en todos los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

1) La Mar Estaba Serena

Diseñe un algoritmo modular que tenga como dato de entrada la estrofa de una canción terminada en un punto y además una vocal. Su algoritmo debe cambiar todas las vocales de la estrofa por la vocal ingresada y todas las letras en minúsculas.

Además debe contar cuantas vocales hay en la estrofa y cuantas mayúsculas.

En su solución debe declarar e implementar el procedimiento y la función:

void mostrarCancion(char cancion, char vocal);

int esVocal(char cancion);

Ejemplo:

Si la estrofa de entrada es: “**La Mar Estaba Serena, Serena Estaba La Mar.**” y la vocal es: **e**

La salida es: “**le mer estebe serene, serene estebe le mer.**”

Cantidad de vocales: 16

Cantidad de mayúsculas: 8

2) De la trigonometría

Diseñe y escriba un módulo que le permita obtener una tabla con los valores de los senos de los ángulos comprendidos en el intervalo (0° , 360°) incrementando de a 30 grados. La misma debe consignar los valores del ángulo en grados y radianes y el valor del seno correspondiente.

Para llevar a cabo esta tarea, debe usar la librería de funciones matemáticas.

RECUERDE: 180° equivalen a 1π radián.

Use la constante PI definida en <math.h>

Controle los valores obtenidos con la siguiente tabla:

x[grados]	x[radianes]	seno(x)
0	0	0
30	0,523599	0,5
60	1,047198	0,866025
90	1,570796	1
120	2,094395	0,866025
150	2,617994	0,5
180	3,141593	0
210	3,665191	-0,5
240	4,18879	-0,866025
270	4,712389	-1
300	5,235988	-0,866025
330	5,759587	-0,5
360	6,283185	0

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

3) Autocalificación

La cátedra de Elementos de Computación y Lógica necesita automatizar la corrección de parciales, para esto se requiere que a partir de los puntajes individuales (obtenidos por un estudiante) para cada punto de un parcial, se informe si el estudiante aprueba o desaprueba y además muestre la nota obtenida.

Cada punto de un parcial tiene un puntaje de manera que sumados llegan a 10.

Se aprueba con 6, pero además es requisito cumplir con un porcentaje mínimo de nota para cada punto, tal como se detalla a continuación:

Punto	Puntaje sobre 10	Porcentaje mínimo para dar por válido
1	3,5	40%
2	2,5	30%
3	4	70%

Resuelva implementando al menos un procedimiento o una función.

Datos de prueba:

Alumno1:	Punto1: 2 Salida: Aprobada/o	Punto2: 2,5 Nota: 7,5	Punto3: 3
Alumno2:	Punto1: 3,5 Salida: Desaprobada/o	Punto2: 0,4 Nota: 7,9	Punto3: 4
Alumno3:	Punto1: 1,5 Salida: Desaprobada/o	Punto2: 1 Nota: 5,5	Punto3: 3

4) Años bisiestos

Realizar una función para determinar si un año es bisiesto.

Luego realice un programa que permita ingresar dos años y muestre en pantalla la cantidad de años bisiestos que hay entre los dos años ingresados. Muestre todos los años bisiestos encontrados en el rango.

A tener en cuenta: Son bisiestos todos los años múltiplos de 4, excepto aquellos que son múltiplos de 100; también son bisiestos los múltiplos de 400.

Ejemplos de años bisiestos: 800, 1964, 1984, 2004, 2008, 2012, 2024, ...

Ejemplos de años no bisiestos: 200, 1700, 1800, 1900, 1965, 2100, ...

Datos de prueba: Ingreso [1950,2022]

Salida: La cantidad de años bisiestos entre 1800 y 2022 es: 18

Los Años bisiestos son: 1952, 1956, 1960, 1964, 1968, 1972, 1976, 1980, 1984, 1988, 1992, 1996, 2000, 2004, 2008, 2012, 2016, 2020

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

5) Calculadora Reducida

Se cuenta con una calculadora que tiene un set de instrucciones reducido, las operaciones aritméticas que puede realizar son solo sumas y restas.

Se necesita la ayuda de usted como programador para que diseñe un algoritmo tal que se ingresen dos números y un código de operación para que devuelva el resultado.

Para esto debe diseñar y hacer uso de las siguientes funciones que dados dos números enteros a y b :

- $\text{suma}(a,b)$: retorne su suma.
- $\text{producto}(a,b)$: retorne su producto (multiplicación).
- $\text{resta}(a,b)$: retorne su resta.
- $\text{division}(a,b)$: retorne su división entera. Donde a es el dividendo y b el divisor.
- $\text{resto}(a,b)$: retorne el resto de la división entera.
- $\text{potencia}(a,b)$: retorne a elevado a la b . Es decir: a es la base y b el exponente.

ATENCIÓN: Reutilice las funciones en caso de ser necesario para definir nuevas funciones.

Observaciones:

- Trabajar con números enteros positivos.
- Solo puede utilizar los operadores de suma y resta.

Funciones de biblioteca: `<math.h>` y `<ctype.h>`

Función	Descripción	Ejemplo de uso
<code>sqrt</code>	raíz cuadrada de x	<code>sqrt(900.0)</code> es 30.0
<code>exp(x)</code>	función exponencial e^x	<code>exp(1.0)</code> es 2.718282 <code>exp(2.0)</code> es 7.389056
<code>log(x)</code>	logaritmo natural de x (base e)	<code>log(2.718282)</code> es 1.0 <code>log(7.389056)</code> es 2.0
<code>log10(x)</code>	logaritmo de x (base 10)	<code>log10(1.0)</code> es 0.0 <code>log10(100)</code> es 2.0
<code>ceil(x)</code>	redondea a x al entero más pequeño que no sea menor que x	<code>ceil(9.2)</code> es 9.0 <code>ceil(-9.8)</code> es -9.0
<code>floor(x)</code>	redondea a x al entero más grande no mayor que x	<code>floor(9.2)</code> es 9.0 <code>floor(-9.8)</code> es -9.0
<code>pow(x,y)</code>	x elevado a la potencia y	<code>pow(2.7)</code> es 128.0
<code>sin(x)</code>	seno de x expresado en radianes	<code>sin(0.0)</code> es 0.0
<code>cos(x)</code>	coseno de x expresado en radianes	<code>cos(0.0)</code> es 1.0
<code>tan(x)</code>	tangente de x expresado en radianes	<code>tan(0.0)</code> es 0.0

Función	Descripción
<code>int isdigit (int c)</code>	SI (c es un dígito) ENTONCES regresa un valor verdadero SINO regresa un valor falso
<code>int isalpha (int c)</code>	SI (c es una letra) ENTONCES regresa un valor verdadero SINO regresa un valor falso
<code>int isalnum (int c)</code>	SI (c es un dígito o una letra) ENTONCES regresa un valor verdadero SINO regresa un valor falso
<code>int islower (int c)</code>	SI (c es una letra minúscula) ENTONCES regresa un valor verdadero SINO regresa un valor falso
<code>int isupper (int c)</code>	SI (c es una letra mayúscula) ENTONCES regresa un valor verdadero SINO regresa un valor falso
<code>int tolower (int c)</code>	SI (c es una letra mayúscula) ENTONCES regresa c como una letra minúscula SINO regresa c sin cambios
<code>int toupper (int c)</code>	SI (c es una letra minúscula) ENTONCES regresa c como una letra mayúscula SINO regresa c sin cambios

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Ejercicios Extras (Primera Parte)

TEMA: Selección. Iteración. Funciones



Para el desarrollo de los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.

1) Diptongo

Dada una frase terminada en punto, cuente la cantidad de palabras que contienen diptongos.

Aclaración: Un diptongo se forma cuando se unen dos vocales contiguas en una misma sílaba. Para el análisis de la vocal use una función.

Ejemplo:

Si la frase de entrada es: “El pianista dio un concierto excepcional.”

La salida es: **Cantidad de diptongos: 4**

2) Mínimo Común Múltiplo

A partir de la función que diseñó en el ejercicio 3) del TP1, realice un algoritmo que calcule el mínimo común múltiplo de 6 pares de números enteros. Recuerde, los números ingresados pueden estar desordenados.

Adicionalmente, el algoritmo debe mostrar la suma de los dígitos que componen el mcm de cada par de números que se evalúa.

Datos de Prueba

número 1	número 2	mcm	suma
24	36	72	9
180	324	1620	9
240	86	10320	6
90	16	720	9
12	82	492	15
30	76	1140	6

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

3) Número Perfecto

Diseñe una función que reciba un número entero positivo y retorne si es perfecto, según corresponda.

Realice un algoritmo que a partir de una secuencia indeterminada de números enteros, diga si se trata de números perfectos y cuente la cantidad de estos. Realice los controles que crea necesarios.

Aclaración: Un número es perfecto cuando es igual a la suma de sus divisores excepto él mismo. Números perfectos: 6, 28, 496, 8.128.

$$6 = 3 + 2 + 1$$

$$28 = 1 + 2 + 4 + 7 + 14$$

4) Campaña de prevención

Hipertensión es el término que se utiliza para describir la presión arterial alta. Siendo el principal factor de riesgo para sufrir una enfermedad cardiovascular, el servicio de cardiología del Hospital Centro de Salud Zenón J. Santillán organiza una jornada de concientización sobre el control de la misma en la población adulta.

Se desea registrar la presión arterial de un número indeterminado de personas a lo largo de un día. Esta se expresa en dos números, por ejemplo 112/78 mm Hg. El primer número (llamado presión sistólica), es la presión cuando late el corazón. El segundo número (llamado presión diastólica) es la presión cuando el corazón descansa entre latidos.

Mostrar un informe sobre los pacientes evaluados indicando el porcentaje por cada categoría según la siguiente tabla:

CATEGORÍA DE LA PRESIÓN ARTERIAL	SISTÓLICA mm Hg (número de arriba)		DIASTÓLICA mm Hg (número de abajo)
NORMAL	MENOS DE 120	y	MENOS DE 80
ELEVADA	120-129	y	MENOS DE 80
PRESIÓN ARTERIAL ALTA (HIPERTENSIÓN) NIVEL 1	130-139	o	80-89
PRESIÓN ARTERIAL ALTA (HIPERTENSIÓN) NIVEL 2	140 O MÁS ALTA	o	90 O MÁS ALTA
CRISIS DE HIPERTENSIÓN (consulte a su médico de inmediato)	MÁS ALTA DE 180	y/o	MÁS ALTA DE 120

Además, mostrar el promedio de edades para cada categoría y si la presión arterial alta de Nivel 1 se registró más en varones o en mujeres.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Datos de Prueba

Edad	Género (F/M)	Sistólica	Diastólica	Categoría
45	F	118	70	Normal
37	M	115	68	Normal
60	M	120	75	Elevada
66	F	130	80	Nivel 1
72	M	130	85	Nivel 1
63	F	132	80	Nivel 1

Total registrado: 6 personas

Mujeres en Nivel 1: 2

Varones en Nivel 1: 1

SALIDA DEL PROGRAMA

Categoría	% por Cat.	Prom. edad
Normal	33.33	41
Elevado	16.66	60
Nivel 1	50	67
Nivel 2	0	0
Crisis	0	0

La presión arterial de Nivel 1 es más frecuente en mujeres.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 6

TEMA: Arreglos

“Un arreglo es una colección de variables ordenadas e indexadas, todas de idéntico tipo que se referencian usando un nombre común”.

Funciones de biblioteca del archivo de cabecera <string.h>

Las variables cad1 y cad2 son arreglos de caracteres.

int strlen(cad1)	Retorna la longitud de cad1	
int strcmp(cad1, cad2)	Compara cad1 con cad2, carácter a carácter	SI (cad1[i] < cad2 [i]) ENTONCES Retorna entero < 0 SI (cad1[i] = cad2 [i]) ENTONCES Retorna 0 SI (cad1[i] > cad2 [i]) ENTONCES Retorna entero > 0
int strncmp(cad1,cad2,n)	Compara hasta n caracteres de la cad1 con cad2 sin diferenciar mayúsculas de minúscula.	SI (cad1[i] < cad2 [i]) ENTONCES Retorna entero < 0 SI (cad1[i] = cad2 [i]) ENTONCES Retorna 0 SI (cad1[i] > cad2 [i]) ENTONCES Retorna entero > 0
char *strcpy(cad1, cad2)	Copia cad2 a cad1, incluyendo el terminador “\0”. Retorna cad1	
char *strncpy(cad1, cad2,n)	Copia hasta n caracteres de la cad2 a cad1. Retorna cad1. Rellena con “\0” si cad2 tiene menos de n caracteres	
char *strcat(cad1, cad2)	Concatena la cad2 al final de cad1. Retorna cad1	



Para el desarrollo del trabajo práctico, en todos los ejercicios, diseñar y escribir un algoritmo que resuelva la consigna propuesta. Luego, codificar en Lenguaje C. Realice pruebas para distintos conjuntos de datos y asigne convenientemente el tipo de los mismos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

1. Arreglo con números

- Declare un arreglo de números enteros **arreNumero** y cargue con 20 números enteros aleatorios comprendidos en el intervalo [0,400]. Luego calcule y muestre por pantalla la suma de los valores que hay en posiciones pares del mismo (0,2,...) y cuente cuántos números pares hay en las posiciones impares (1,3,...).
- Modifique el programa anterior para que el número de valores aleatorios no sea fijo sino que se lea como entrada y sea como máximo 20. ¿Qué ocurre si indicamos más de 20? Lea también los valores para conformar el intervalo en consideración, realice los controles convenientes.

Para trabajar declare y defina las siguientes funciones:

```
void cargarArreglo(int arreglo[], int tama, int num1, int num2)
void mostrarArreglo(int arreglo[], int tama)
void sumarycontarArreglo(int arreglo[], int tama)
```

2. Arreglos con caracteres

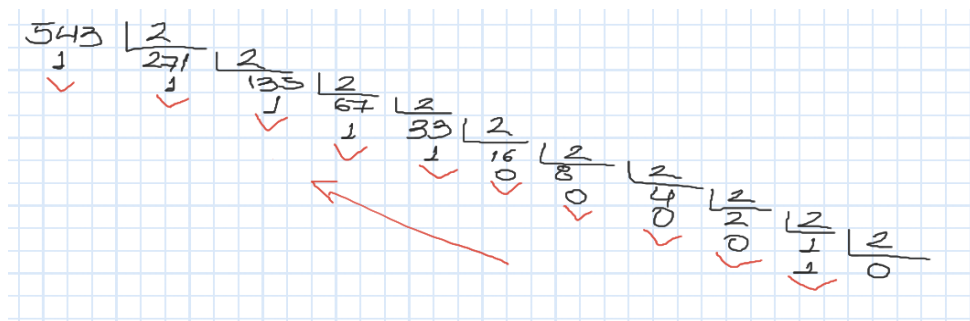
- Dado el nombre y el apellido de una persona, por separado, unirlos en una única cadena de manera que quede de la forma: "Apellido, Nombre". No use la función **strcat()**.
- A partir del nombre completo de una persona (al menos un apellido y dos nombres), compactar la frase en el mismo arreglo, es decir, eliminar los espacios en blanco. Además, sume los equivalentes en código ascii de las letras mayúsculas. Por ejemplo:

Entrada: "Héctor Eduardo Reglero Montaner"

Salida: "HéctorEduardoRegleroMontaner" - Suma: 300 (72+69+82+77)

3. Conversor de Números

El sistema de numeración binario utiliza sólo dos dígitos: 0,1 para representar números. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Para convertir un número decimal a base 2 se realizan divisiones sucesivas en dos de manera que los restos forman el equivalente en binario.



De esta forma $(543)_{10} = (100001111)_2$

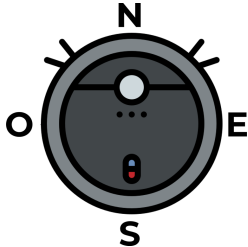
La tarea consiste en leer un número decimal y mostrar su equivalente binario. Luego de almacenar los restos, use un arreglo auxiliar para guardar correctamente el número en base 2.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

4. Aspirabot

Se tiene una aspiradora robot que puede estar mirando en una de 4 direcciones posibles: norte, sur, este y oeste. A continuación se muestra una mirando hacia el norte, vista desde arriba.



Recibe una lista de instrucciones, de dos tipos:

- **Girar a la izquierda**, en sentido antihorario visto desde arriba:
norte → oeste → sur → este → norte
- **Girar a la derecha**, en sentido horario visto desde arriba:
norte → este → sur → oeste → norte

Comienza mirando hacia el norte. El objetivo final es que quede mirando al sur. Por ejemplo, si un robot que mira hacia el norte recibe una orden de girar a la derecha (en sentido horario), quedaría mirando hacia el este.

El robot va ejecutando todas las instrucciones una por una, y el proceso termina cuando el robot queda mirando al sur. No importa si luego existen más instrucciones.

Se desea conocer cuantas instrucciones hay que procesar hasta que el robot quede mirando al sur, podría nunca suceder. Escribir una función que calcule este dato a partir de la cadena de instrucciones.

Entrada: HAHAHHHAAHA - Salida: 8

Entrada: HAA - Salida: -1

Nota: Una H indica un giro en sentido horario, y una A un giro en sentido antihorario.

5. Adivinando la Palabra

Programar un prototipo para un juego llamado Adivinando la Palabra. El juego funciona así:

- Primero se debe ingresar una palabra. Luego según la cantidad de caracteres, se debe mostrar en pantalla guiones incógnita reemplazando los caracteres.
- Un usuario tiene que adivinar la palabra ingresando alguna letra.
 - Si la letra se encuentra en la palabra, se debe volver a mostrar en pantalla los guiones incógnita, salvo las letras encontradas por el usuario.
 - Si el usuario ingresa una letra que no está en la palabra se debe descontar un intento. Y tiene un máximo de 3 intentos para adivinar la palabra.

Si el participante gana se debe mostrar un mensaje informando que adivinó la palabra o si perdió, informar que agotó los tres intentos.

Ejemplo:

- 1- Ingresa palabra: **programacion**
- 2- Muestra en pantalla: - - - - -
- 3- Ingresa una letra: **a**
- 4- Muestra en pantalla: - - - - a - a - - - Intentos: **3**
- 5- Ingresa una letra: **e**
- 6- Muestra en pantalla: - - - - a - a - - - Intentos: **2**

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 7

TEMA: Punteros y Arreglos

“Un puntero es una variable que contiene la dirección de una variable”. Kernighan & Ritchie

CONSIGNA GENERAL



Para los desplazamientos en los arreglos, use exclusivamente notación de punteros.



Para el desarrollo de los puntos 1 y 2 realice un seguimiento ordenado y a conciencia de los pasos indicados.

Problema 1: Calentando Motores

Objetivo: entender la implementación de punteros sin uso de arreglos

01. Escriba un programa que lea dos números enteros y los almacene en las variables **a** y **b**. Muestre en pantalla los valores leídos en **a** y **b**.
02. Cree dos punteros enteros llamados **p** y **q**. Asigne en los punteros **p** y **q** la dirección de memoria de **a** y **b** respectivamente.
03. Muestre el contenido de **p** y **q**.
04. Muestre el contenido apuntado por **p** y **q**.
05. Muestre las direcciones de memoria de **p** y **q**.
06. Lea un número y guárdelo en una variable llamada **c**. Muestre el contenido de **c** y la dirección de memoria de la variable **c**.
07. Asigne a la variable **c** el valor de **a** utilizando el puntero **p**. Muestre las variables **a** y **c**.
08. En la variable **c** guarde la suma de **a+b** utilizando los punteros **p** y **q**. Muestre las variables **a**, **b** y **c**.
09. ¿Qué sucede si ejecutamos ***p=*p+1**?
10. ¿Qué sucede si ejecutamos **p=p+1**?
11. Dada la siguiente asignación: **p=&a**:
 - ¿Las sentencias ***p=*p+1** y ***p=a+1** son equivalentes?
 - ¿Las sentencias **p=p+1** y **p=&a+1** son equivalentes?

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Problema 2: Sumando Arreglos

El objetivo de este punto es que sirva de guía en la implementación de punteros y arreglos en lenguaje C.

Lea una línea de texto en minúsculas con la función `gets`, almacene en un arreglo de caracteres de nombre `cad` [100]. Luego, cuente las consonantes presentes y convierta a mayúsculas.

01. En el código del ejercicio anterior, declare en forma local a `main` una variable puntero **punt**, que apunte a un objeto de tipo `char`.
02. Asigne la dirección inicial del arreglo `cad` a la variable puntero **punt**. Haga esta tarea de dos maneras distintas.
03. Muestre en pantalla la dirección de la variable puntero **punt**. Use el indicador de formato apropiado. Muestre también la dirección de la primera componente del arreglo **cad**. Acceda a la misma de forma indirecta, o sea, a través de la dirección.
04. Muestre en pantalla el contenido de la quinta celda del arreglo **cad**. Refiérase a ella utilizando: notación de subíndice del arreglo y notación de puntero con desplazamiento mediante **punt**.
05. Suponiendo que **punt** apunta al principio del arreglo **cad**, ¿cuál es la dirección referenciada por `punt+3`? ¿Cuál es el valor almacenado en esa dirección?
06. Escriba las instrucciones necesarias para visualizar en la pantalla cada uno de las componentes del vector **cad**, utilizando notación de punteros con el puntero **punt**.
07. Muestre en pantalla los elementos del arreglo **cad**, mediante subíndices del puntero **punt**.

Problema 3: Analizando el Curso

Diseñe un algoritmo que le permita obtener de un arreglo numérico que contiene los promedios de los alumnos de un curso:

- Los 3 mejores promedios, indicando el orden.
- El promedio más bajo del curso.
- El promedio general del curso.
- La cantidad de alumnos que superan el promedio general.

Implemente en C. Realice funciones para cargar y mostrar los datos del arreglo.

Problema 4: Mezclando Cadenas

Diseñe un algoritmo que lea dos cadenas de caracteres alfabéticos y un número. La tarea consiste en insertar **la segunda cadena en la primera**, a partir del carácter de la primera secuencia que está en la posición indicada por el número.

Finalmente imprimir **la primera cadena**. Implemente en C.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Ejemplo:

Cadena 1: "Octubre" - **Cadena 2:** "Verano" - **Número:** 5

Cadena 1 de Salida: **OctuVeranobre**

Nota: Usar notación de punteros. Realizar los controles que sean necesarios. No usar la función concatenar de la biblioteca string.h

Problema 5: Nueva documentación

Un documento nacional de identidad (DNI), es un documento emitido por una autoridad administrativa competente para permitir la identificación personal de los ciudadanos y su posesión es obligatoria en la mayoría de los países hispanos.

En España, también se le denomina carné de identidad. Se trata de una tarjeta plastificada o de policarbonato donde se detalla el nombre y apellidos del titular, fecha de nacimiento, dirección, progenitores, sexo, dirección de residencia, localidad y provincia de nacimiento, y contiene una fotografía y un número de identificación formado por 8 cifras más una letra de control. A esta combinación de números y letra se le denomina número de identificación fiscal (NIF), que sería el equivalente al CUIT (Código de Identificación Tributaria) que se usa en nuestro país. Suponga Ud. que las autoridades del Registro Nacional de las Personas de nuestro País, le han solicitado la tarea de escribir un programa que permita otorgar a todos los ciudadanos argentinos lo que los españoles llaman NIF, como complemento de identificación fiscal. Por ejemplo: si el DNI es 30597721 el NIF correspondiente es: Q30597721.

Para determinar la letra se usa una tabla que contiene 23 letras mayúsculas; se puede determinar la posición de la letra en la tabla mediante la siguiente expresión: $\text{DNI} \% 23$.

La tabla es: | T | R | W | A | G | M | Y | F | P | D | X | B | N | J | Z | S | Q | V | H | L | C | K | E |

Considere que en nuestro País los D.N.I. comienzan su numeración a partir de 10.000.000 (diez millones). Averigüe Ud. Cuál es el orden de los últimos DNI entregados.

Diseñe y escriba un algoritmo de pocas líneas que le permita indicar el NIF a un conjunto de N ciudadanos. Codifique luego en Código C y pruebe para distintos conjuntos de datos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 8

TEMA: Mecanismos de Pasaje de Parámetros - Asignación de memoria dinámica



*Para los desplazamientos en los arreglos, preferentemente notación de punteros.
Siempre que sea posible declare los arreglos de forma dinámica.*

Problema 1: Cadenas apuntadas

Implemente un programa que realice las siguientes operaciones en una cadena:

- a. **Función Buscar:** operación que le permite determinar si la cadena contiene un determinado carácter. La función debe devolver un valor de verdad y además, la cantidad de veces que el carácter aparece. Use el siguiente prototipo:

int buscar(char *cad, char c, int *cant);

- b. **Función Reemplazar:** permite reemplazar un carácter en una determinada posición de la cadena. Use el siguiente prototipo:

void reemplazar (char *cad, int pos, char c);

- c. **Función Insertar:** genera una nueva cadena insertando un carácter determinado después de la aparición del mismo carácter en la cadena original.
Esta nueva cadena debe generarse de forma dinámica. Reutilice la función buscar de ser necesario.

Use el siguiente prototipo: **void insertar(char *cad, char *nueva, char c);**

Ejemplo:

Frase original: "aprender programacion es muy facil"

Carácter a insertar: **a**

Nueva frase: "aaprender prograamaacion es muy faacil"

Problema 2: Cadena procesada

Escribir un **único** módulo que dada una frase, devuelva los siguientes datos:

- La longitud de una cadena (sin utilizar la función strlen).
- La cantidad de vocales.
- La cantidad de palabras.

Problema 3: Tiro al blanco

a) En una Kermés se presenta el juego de tiro al blanco con dardos. El costo de un tiro es de \$100. Los premios según la puntería son:

- El círculo rojo con valor 10 tiene un premio de \$500.
- El círculo de valor 9 tiene un premio de \$200.
- El círculo de valor 8 recupera los \$100.
- El círculo de valor 7 recupera \$50.
- El resto de círculos no tiene premio.



PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Realice un programa que permita ingresar cuantos tiros va a realizar el jugador. Luego, simule los tiros. Por último, muestre cuánta plata gastó, cuánta plata ganó y el saldo final del jugador.

Utilice las siguientes funciones:

void cargarTiros(int *puntos, int tiros); //Función para cargar los puntajes de los tiros.
int calcularGanancia(int *puntos, int tiros); //Función que calcula cuánto ganó el jugador.

b) Modifique el programa anterior para que funcione con un número indeterminado de usuarios. Además de mostrar los resultados de cada jugador, al finalizar el juego para todos los usuarios, muestre el monto total pagado por los mismos, el monto total en premios y el saldo final. Elija la marca final que considere apropiada.

Para esto modifique la función **calcularGanancia** con el siguiente prototipo:

void calcularGanancia(int *puntos, int tiros, int *totalPagos, int *totalPremios)

donde:

- **totalPagos:** acumula el pago efectuado por los jugadores
- **totalPremios:** acumula la cantidad de premios ganados por los jugadores

Problema 4: Las notas del profe

a) Realice un programa que permita calcular el promedio de notas de los alumnos de un curso. La cantidad de alumnos dependerá del profesor que cargue las notas (según la cantidad de alumnos que posea en su materia).

Al iniciar el programa debe preguntar cuántos alumnos tiene. Simule las notas con la función `rand()` con números aleatorios de 0 a 10 y con 2 cifras decimales.

Calcule y muestre el promedio del curso y la mejor nota.

Utilice los prototipos de las siguientes funciones:

void cargarNotas(float *notas, int alumnos) // Función para simular las notas
float calcularPromedio(float *notas, int alumnos, float *promedio); // Función que calcula y guarda el promedio, además retorna la mejor nota.

b) Modifique el programa de manera tal que permita cargar las notas de los alumnos de las materias: (0)Cálculo, (1)Álgebra, (2)Programación. Cada materia puede tener diferente cantidad de alumnos. Muestre el promedio de notas y la nota más alta de cada materia. Por último muestre la materia con mejor promedio y la materia que tiene la mayor nota.

Nota: cree arreglos dinámicos para las notas de cada materia.

Guarde en un arreglo las notas promedio y en otro arreglo la mayor nota de cada materia.

Trate de analizar y utilizar las siguientes funciones:

void cargarNotas(float *notas, int alumnos);
void calcularPromedio(float *notas, int alumnos, float *promedio, float *mayorNota);
void mejores(float *promedio, float *mayorNota, int materias);

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 9 TEMA: Estructuras (Primera Parte)



Trabaje de forma ordenada en cada ejercicio. Identifique entre la información que se brinda, el tipo de dato en cada caso y la presencia de estructuras que se puedan anidar.

Problema 1: Se necesita almacenar información para gestionar la biblioteca de la FACET.

01. A partir de la siguiente información sobre las publicaciones disponibles, organice la misma de forma estructurada.

- Título.
- Editorial.
- Edición. (1,2,...)
- Fecha de publicación.
- Autor (Apellido y Nombre)
- ISBN. (número de 10-13 dígitos)
- Categoría. (Libro: L - Revista: R - Tesis: T)
- Cantidad de ejemplares.

02. Utilice typedef para definir el tipo de dato fecha y autor. Úselos en la estructura principal donde organiza la información de las publicaciones.

03. Escriba un módulo que permita **cargar** en el sistema los datos de una publicación.

04. Escriba un módulo que **muestre** la información de una publicación cualquiera.

05. Escriba un programa para la prueba de los módulos anteriores, cargue y muestre una publicación.

06. Luego, declare un arreglo del tipo de dato definido en el inciso 01 y carguelo con 5 publicaciones, finalmente muestre las mismas.

Problema 2: La Coordinación de Educación Digital del Ministerio de Educación de la Provincia tiene a su cargo la revisión digital de la información que las diferentes Direcciones y Modalidades producen para luego ser publicados como artículos en un repositorio de recursos online destinado a la comunidad educativa en general. Para llevar un orden del material se registran en un planilla los siguiente datos:

- Nombre del artículo
- Destinatarios (Alumnos, Docentes, Directivos)
- Nivel: (Inicial, Primario, Secundario, Superior)
- Área: (Lengua, Matemática, Ciencias Naturales, Ciencias Sociales, Robótica, Teatro, Música, etc.)
- Descripción: (resumen del tema del artículo)
- Datos de quien controla el artículo: (Apellido y Nombre)
- Fecha de control
- Estado: (0: no controlado, 1: controlado, 2: publicado)

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

01. Organice la información de forma estructurada.
02. Escriba un módulo que permita **cargar** en el sistema la información de N artículos.
03. Escriba un módulo que **muestre** todos los artículos existentes.
04. Escriba un módulo que modifique el estado de todos los artículos que coinciden con nivel, área y destinatarios que recibe como parámetros. Indique también como información el valor que debe asumir el estado, es decir, 0,1 o 2.
05. En el programa principal declare un arreglo del tipo definido en el inciso 01, cargue con N artículos y luego modifique el estado de aquellos que cumplan con los criterios nivel, área y destinatarios ingresados por teclado.

Problema 3: Se desea gestionar la información de los clientes de un supermercado mayorista (apellido, nombre, dni, fecha de nacimiento, y datos de la cuenta que son el código numérico de cliente y el saldo que debe). Defina una estructura para guardar los datos de un cliente en donde la fecha de nacimiento y los datos de la cuenta son de tipo struct. Cargue toda la información de 5 clientes.

Realice un programa que inicialmente muestre un listado de los clientes con apellido, nombre y código. Luego, presentar el siguiente menú:

01. Aumentar el saldo deudor (debe sumar el nuevo valor a la cuenta del cliente).
02. Disminuir saldo deudor: Si al restar el saldo deudor la cuenta del cliente queda saldada mostrar el mensaje: "Cuenta saldada, no tiene deuda", de lo contrario mostrar el valor actualizado.
03. Salir

Si la opción ingresada es 1 o 2 pedir el código del cliente al cual se aplica la operación y el monto.

El programa debe finalizar con la opción salir, con cualquier otra opción debe retornar siempre al menú principal.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 10

TEMA: Arreglos de Estructuras, Funciones, Punteros



Trabaje de forma ordenada en cada ejercicio. Identifique entre la información que se brinda, el tipo de dato en cada caso y la presencia de estructuras que se puedan anidar. **Use pasaje de parámetros por referencia para los arreglos. En los puntos 2 y 3 use asignación dinámica de memoria.**

Problema 1: Retome el punto 1 del trabajo práctico N° 9. Realice los ajustes que crea necesarios.

- ★ A partir del tipo base predefinido en el punto 01, declare un arreglo de estructuras de nombre **INVENTARIO** que permita almacenar la información de **n** publicaciones.
- ★ Escriba un módulo que permita cargar el arreglo completo.
- ★ Escriba un módulo que muestre todas las publicaciones.
- ★ Escriba un módulo que modifique la categoría de todas publicaciones según:
 - Si tiene ISBN corresponde Libro
 - Si su ISBN es 0 y la editorial es FACET se trata de una Tesis.
 - Si su ISBN es 0 y la editorial es distinta a FACET es Revista.
- ★ Escriba un módulo que a partir del ISBN como dato, reste o sume la cantidad de ejemplares de una publicación. Enviar como parámetros la información necesaria y realizar los controles pertinentes.

Problema 2: Se desea gestionar la información de los clientes de un banco:

- Apellido
- Nombre
- Dni
- Fecha de nacimiento
- Contraseña
- Datos de la cuenta bancaria: (CBU y Saldo)

01. Organice la información de forma estructurada.

02. Escriba un módulo que permita **cargar** en el sistema la información de N clientes.

03. Escriba un módulo que **muestre** todos los clientes.

04. Luego de cargar la información de los clientes debe mostrarse un menú para operar:

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Menú del programa:

1. Mostrar datos de los clientes (muestra todos los datos de los clientes y sus cuentas).
2. Seleccionar cliente para operar: se solicita que ingrese el número de cliente (el orden en el que se encuentra) para operar con el siguiente menú:
 1. Cambiar contraseña: permite ingresar una nueva contraseña.
 2. Depositar un monto: debe sumar el valor a la cuenta del cliente.
 3. Retirar dinero: resta el dinero siempre y cuando tenga el saldo disponible.
 4. Transferir dinero a otro cliente (debe solicitar el número de cliente a transferir y se debe verificar que tenga el saldo disponible para transferir).
 5. Consultar Saldo
 6. Mostrar datos de la cuenta: Muestra el nombre del cliente y el CBU.
 7. Volver al menú principal.
3. Salir del programa

```
=====MENU PRINCIPAL=====
Ingrese una opcion:
1-Mostrar Clientes
2-Seleccionar Cliente para operar
3-Salir
```

Al seleccionar la opción 2:

```
Ingrese numero de cliente
2
=====MENU DEL CLIENTE [2]=====
Ingrese una opcion:
1-Cambiar Contraseña
2-Depositar dinero
3-Retirar dinero
4-Transferir dinero
5-Consultar saldo
6-Mostrar datos de la cuenta
7-Volver
```

Problema 3: Plantar nuevos árboles puede ayudar a reducir la concentración de CO₂ en la atmósfera. Los gases de efecto invernadero como el dióxido de carbono y el metano contribuyen significativamente a un clima cambiante. Los bosques son un efectivo sumidero de carbono natural que absorbe gran parte del carbono emitido por la quema de combustibles fósiles. Revertir

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

la deforestación global es un elemento clave de una estrategia de mitigación efectiva para combatir el calentamiento global.

Supongamos que tenemos una empresa de reforestación y queremos hacer un programa que almacene los diferentes tipos de árboles con los que se cuenta. La información relevante es:

- Datos de la especie:
 - Especie o nombre científico (única para cada una)
 - Categoría (caduca, perenne, conífera, frutal)
 - Nombre vulgar.
 - Árboles de hoja caduca: Arce rojo, Aromo, Acacia espinosa, Castaño de Indias.
 - Árboles de hoja perenne: Encina, Alcornoque, Olivo.
 - Árboles de coníferas: Abeto, Pino, Araucaria.
 - Árboles frutales: Naranja, Mango, Limonero, Cerezo, Manzano.
- Clima (tropical, seco, templado, continental y polar)
- Cantidad disponible en el almacén.
- Reponer (indica si se debe reponer ejemplares, por defecto falso)

A partir de la información detallada:

01. Organice la información de forma estructurada.
02. A partir del tipo base predefinido en el punto 01, declare un arreglo de estructuras para almacenar la información de n árboles:
 - a. Escriba un módulo para cargar el arreglo con las distintas especies de árboles.
 - b. Escriba un módulo para mostrar la información de todos los árboles.
03. Escriba un módulo que muestre los datos de la especie de todos los árboles disponibles para una categoría dada, siempre y cuando tenga al menos 1 unidad.
04. Escriba un módulo que actualice el dato Reponer de un ejemplar según la cantidad disponible, si es menor a 5 unidades, se debe reponer. Además, debe informar la cantidad de registros actualizados.