

Trabajo Práctico N° 1

Tema: Condicionales

Fecha de Entrega: 09-04-2019

Repositorio del Trabajo:

<https://github.com/AndresNakanishi/CAECE-LESYN/tree/master/FirstQuarter/informatics/exercices/TP1>

1 Introducción a la Informática

Consignas:

Desarrollar para cada ejercicio un programa Pascal que resuelva el problema planteado, presentar impresos:

- a) Enunciado
- b) Programa
- c) Dos juegos de datos y sus respectivas salidas

Además traer el programa ejecutable.

En algunos casos debe determinar los datos que debe ingresar al proceso .

Cabe **destacar** que el funcionamiento de cada algoritmo **incluye distintos datos que son en pos de mejorar la usabilidad de los mismos**, pero **NO son agregados a los datos de salida**, ya que en muchas consignas **SOLO es requerido el resultado final**.

- 1) En un concurso de salto hípico los jinetes saltan con su caballo, registrándose:

- **Altura**
- **Distancia**

Se determina la siguiente clasificación:

- Bueno el salto **donde la altura y la distancia** superan los **2.5 mt (ambas)**.
- Normal el salto donde altura o distancia superen los **2.5 mt**.
- Malo **cualquier otra marca**.

Determine los datos que se requieren para clasificar un salto y deben ingresarse al proceso.

DATOS DE ENTRADA:

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

2 Introducción a la Informática

Set de Datos 1: Altura => 1.5 | Distancia => 2.6

Set de Datos 2: Altura => 1.6 | Distancia => 1.3

DATOS DE SALIDA:

Set de Datos 1: => Salto Normal

Set de Datos 2: => Salto Malo

Program SaltoHipico;

Uses

crt;

Var

// H => Height (Altura)

// D => Distance (Distancia)

h,d: real;

begin

clrscr;

writeln('Ingrese la altura del salto: ');

readln(h);

clrscr;

writeln('Ingrese la distancia del salto: ');

readln(d);

clrscr;

if((d > 2.5) AND (h > 2.5)) then

// El caso donde tanto la distancia como la altura superan los

2.5m

begin

clrscr;

writeln('Salto Bueno');

end

else if((d > 2.5) or (h > 2.5)) then

// El caso donde la distancia o la altura superan los 2.5m

begin

clrscr;

writeln('Salto Normal');

end

else

// El caso donde ni la distancia o la altura superan los 2.5m

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

3 Introducción a la Informática

```
begin
  clrscr;
  writeln('Salto Malo');
  end;
  writeln('El salto fue de ',h:4:2,' metros y la distancia de ',d:4:2,'
metros');
  readln;
END.
```

2) Un acopiador paga por tonelada según el cereal:

- **Trigo \$ 38**
- **Girasol \$ 56**
- **Soja \$ 67**

Si el porcentaje de **humedad es mayor al 10%**, el importe abonado **sufre una quita del 8%**.

Determine los datos que se requieren para establecer el importe a abonar por un productor que entrega una determinada cantidad de un cereal, con la respectiva medición de humedad.

DATOS DE ENTRADA:

Set de Datos 1: Tipo de Cereal => T | Toneladas => 100 | Humedad => 51

Set de Datos 2: Tipo de Cereal => S | Toneladas => 100 | Humedad => 50

DATOS DE SALIDA:

Set de Datos 1: => \$3496

Set de Datos 2: => \$6164

```
Program Farmer;
```

```
Uses
  crt;
```

```
Const
  t = 38.00;
  s = 67.00;
  g = 56.00;
```

```
Var
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

4

Introducción a la Informática

```
// Sel = Selección
sel: char;
// Quantity => Cantidad
// Humidity => Humedad
// Res => Resultado (Subtotal) = Cantidad * Precio
// Des => Descuento por Humedad
quantity, humidity, res, des: real;
```

BEGIN

```
clrscr;
writeln('Ingrese el tipo de cosecha: ');
writeln('Posibles entradas T => Trigo, G => Girasol, S => Soja');
readln(sel);
sel:= UPCASE(sel);
clrscr;
writeln('Las toneladas de la cosecha: ');
readln(quantity);
clrscr;
writeln('Ingrese el porcentaje de humedad: (1 - 100) ');
readln(humidity);
clrscr;
case sel of
    'G':
        begin
            res:= (g * quantity);
        end;
    'S':
        begin
            res:= (s * quantity);
        end;
    'T':
        begin
            res:= (t * quantity);
        end;
    else writeln('Entrada Incorrecta');
end;

if(humidity >= 10) then
    begin
        des:= (res * 0.08);
    end
else
    begin
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

5 Introducción a la Informática

```
        des:= 0;
    end;
    writeln('Subtotal: $ ',res:5:2);
    writeln('Descuento por humedad: $', des:5:2);
    writeln('Total: $', (res-des):5:2);
    readln;
END.
```

3) Conociendo el peso de una persona (**nro. real**), **informar el estado de la misma de acuerdo con el siguiente criterio:**

- **peso < 40 'DELGADA'**
- **$40 \leq \text{peso} < 60$ 'NORMAL'**
- **$60 \leq \text{peso} < 80$ 'SOBREPESO'**
- **$80 \leq \text{peso}$ 'OBESA'**

Además informar y determinar la talla que le corresponde según las siguientes condiciones:

- **Talle 1 peso ≤ 50**
- **Talle 2 $50 < \text{peso} \leq 80$**
- **Talle 3 $80 < \text{peso}$**

DATOS DE ENTRADA:

Set de Datos 1: Peso => 50

Set de Datos 2: Peso => 100

DATOS DE SALIDA:

Set de Datos 1: => Persona de peso normal de Talle 1

Set de Datos 2: => Persona obesa de Talle 3

```
Program peso;

uses crt;

var
    // Category => Categoría
    // Size => Talla
    category, size: string;
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

6

Introducción a la Informática

```
// Weight => Peso
weight: real;

BEGIN
  clrscr;
  writeln('Ingrese su peso en Kilos: ');
  readln(weight);
  clrscr;
  if(weight < 40) then
    // Caso donde La persona pesa menos de 40 Kilos
    begin
      category:= 'Persona Delgada';
      size:= 'Talle 1';
    end
  else if((weight >= 40) and (weight < 60)) then
    // Caso donde La persona pesa entre de 40 Kilos y menos de 60
    Kilos
    begin
      category:= 'Persona de peso Normal';
      // Volvemos a usar un solo condicional para verificar el
      talle de la persona.
      if (weight <= 50) then
        begin
          size:= 'Talle 1';
        end
      else
        begin
          size:= 'Talle 2';
        end;
      end
    end
  else if((weight >= 60) and (weight < 80)) then
    // Caso donde La persona pesa entre de 60 Kilos y menos de 80
    Kilos
    begin
      category:= 'Persona con Sobrepeso';
      size:= 'Talle 2';
    end
  else
    begin
      category:= 'Persona Obesa';
      size:= 'Talle 3';
    end
  end
end
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

7

Introducción a la Informática

```
end;  
writeln(category,' de ',size);  
readln();  
END.
```

4) Una empresa mayorista aplica descuentos progresivos en tramos, de acuerdo a cantidad de unidades compradas:

- Más de 100 y hasta 500 unidades 5%.
- Más de 500 y hasta 2000 unidades 7%.
- Más de 2000 y hasta 10.000 unidades 10%.
- Más de 10.000 15%.

Se pide para un determinado precio y cantidad **informe el importe a abonar**.

DATOS DE ENTRADA:

Set de Datos 1: Precio => 10 | Cantidad => 650

Set de Datos 2: Precio => 30 | Cantidad => 10.001

DATOS DE SALIDA:

Set de Datos 1: => 6045

Set de Datos 2: => 255025.50

```
Program precio;  
  
uses crt;  
  
var  
    // Quantity => Cantidad  
    quantity: integer;  
    // Price => Precio  
    // Res => Resultado (Subtotal)  
    // Des => Descuento (Descuento)  
    price, res, des: real;  
  
BEGIN  
    clrscr;  
    writeln('Ingrese el precio unitario: ');  
    readln(price);
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

8

Introducción a la Informática

```
clrscr;
writeln('Ingrese la cantidad de articulos vendidos: ');
readln(quantity);
clrscr;
// Calculo del Subtotal
res:= price * quantity;
if(quantity < 100) then
    // Caso donde se compraron menos de 100 unidades
    begin
        des:= 0;
    end
else if((quantity >= 100) and (quantity < 500)) then
    // Caso donde se compraron 100 o más unidades y menos de 500
    begin
        des:= res * 0.05;
    end
else if((quantity >= 500) and (quantity < 2000)) then
    // Caso donde se compraron 500 o más unidades y menos de 2.000
    begin
        des:= res * 0.07;
    end
else if((quantity >= 2000) and (quantity < 10000)) then
    // Caso donde se compraron 2.000 o más unidades y menos de
10.000
    begin
        des:= res * 0.10;
    end
else
    // Caso donde se compraron más unidades de 10.000
    begin
        des:= res * 0.15;
    end;
// Solo para recordar Lo ingresado
writeln('Precio Unitario: $', price:5:2);
writeln('Cantidad: ', quantity);
// Mostrar toda La data
writeln('Subtotal: $', res:5:2);
writeln('Descuento: $', des:5:2);
writeln('Total: $', (res-des):5:2);
readln();
END.
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

5) Una empresa transportista clasifica el tamaño de las cargas (son cubos) según **el peso y la medida de un lado**, de la siguiente forma:

- **Grande si el peso es mayor a 10 kg y el lado mide más de 90 cm.**
- **Medianos si pesa mas de 5kg y hasta 10 inclusive**
- **Chico si es menor o igual e 5 kg. Y el lado mide menos de 50 cm.**

Se pide leer las especificaciones de una carga e **informar la clasificación correspondiente, o “no clasifica”**. Además en el caso de que clasifique **informar cuánto debe abonar** teniendo en cuenta que se cobra **\$5, \$3 y \$2.5 por kilo respectivamente** a las categorías mencionadas.

Determine los datos que se requieren para calcular la clasificación y costo de una carga.

DATOS DE ENTRADA:

Set de Datos 1: Peso => 11 | Tamaño => 95

Set de Datos 2: Peso => 15 | Cantidad => 60

DATOS DE SALIDA:

Set de Datos 1: => Costo => 55 | Clasificación => Grande

Set de Datos 2: => Costo => 0 | Clasificación => No clasifica

```
program cajas;

uses crt;

const
    // Constantes del Valor definido en el enunciado para cada tipo
    de caja
        big = 5.00;
        medium = 3.00;
        small = 2.50;

var
    // Tipo de Caja (typeof)
    typeof: string;
    // Peso, Tamaño y Resultado
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
// Comentario sobre res: Solo es para aumentar la legibilidad del código
weight, size, res: real;

BEGIN
  clrscr;
  writeln('Ingrese el peso en KG: ');
  readln(weight);
  clrscr;
  writeln('Ingrese el tamaño de la caja en CM: ');
  readln(size);
  clrscr;
  if((weight < 5) and (size < 50)) then
    // Caja pequeña, con un peso menor a 5Kg y una longitud menor a 50cm.
    begin
      typeof:= 'Chica';
      res:= small * weight;
    end
  else if((weight > 10) and (size > 90)) then
    // Caja grande, con un peso mayor a 10Kg y una longitud superior a 90cm.
    begin
      typeof:= 'Grande';
      res:= big * weight;
    end
  else if((weight >= 5) and (weight <= 10)) then
    // Caja mediana, con un peso mayor a 5Kg hasta 10kg inclusive y su longitud no importa.
    // Problema del cliente: Olvido de definir qué pasa con la longitud de las cajas en clasificación "mediana"
    begin
      typeof:= 'Mediana';
      res:= medium * weight;
    end
  else
    // Todo lo que escape a las especificaciones del cliente,
    // se incluirá en esta sección hasta la próxima revisión/reunión con el cliente
    // donde NO se clasificará la caja ingresada
    begin
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

        typeof:= 'No clasifica.';
        res:= 0;
    end;
    writeln('Tipo de Caja: ',typeof);
    writeln('Precio Total: $',res:5:2);
    readln();
END.

```

6) El arancel que cobra una institución universitaria depende de la cantidad de materias que se cursan. Por cada una **se abonan X pesos**, pero si son más de 3 recibe **un descuento de Z pesos por cada materia por arriba de 3. La inscripción comienza el día 1**, si se anota **antes del día 5 tiene un descuento del 5%**, si **lo hace después del 25 tiene un recargo de 12%**.

Determine los datos que se requieren para calcular el importe a abonar por un alumno.

DATOS DE ENTRADA:

Set de Datos 1: Día de Inscripción => 4 | Descuento por más de 3 materias => 10 | Valor por Materia => 100 | Cantidad de Materias => 6

Set de Datos 2: Día de Inscripción => 25 | Descuento por más de 3 materias => 20 | Valor por Materia => 300 | Cantidad de Materias => 10

DATOS DE SALIDA:

Set de Datos 1: => Total => 540

Set de Datos 2: => Total => 2940

```

program university;

uses crt;

var
    // Día de La inscripción
    // Cantidad de Materias => subjects
    day, subjects: integer;
    // Precio por Materia => pricePerSubject
    // Descuento por más de 3 materias => discount
    // Descuento por inscripción sobre el TOTAL => inscriptionP
    // Sobrecargo por inscripción sobre el TOTAL => charge

```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
// Resultado => Subtotal
pricePerSubject, discount, inscriptionP, charge, res: real;

BEGIN
  clrscr;
  writeln('Ingrese el dia de inscripcion: 01 - 31');
  readln(day);
  clrscr;
  writeln('Ingrese el descuento por mas de 3 materias: (1 - 100) ');
  readln(discount);
  clrscr;
  writeln('Ingrese el valor por materia: ');
  readln(pricePerSubject);
  clrscr;
  writeln('Ingrese la cantidad de materias: ');
  readln(subjects);
  clrscr;
  // Calculo el subtotal
  res:= pricePerSubject * subjects;
  // Asumiendo que el usuario ingreso un entero del 1 - 100, se corre
  La coma para facilitar la operatoria
  discount:= discount / 100;

  if(day <= 5) then
    // Según el día de inscripción
    begin
      inscriptionP:= res * 0.05;
    end
  else if(day >= 25) then
    begin
      charge:= res * 0.12;
    end
  else
    begin
      inscriptionP:= 0.00;
    end;
  if (subjects > 3) then
    begin
      discount:= (subjects - 3) * pricePerSubject * discount;
    end;
  writeln('Cantidad de Materias: ',subjects);
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
writeln('Precio por Materia: ', pricePerSubject:5:2);
writeln('Subtotal: ', res:5:2);
writeln('Descuento:', (inscriptionP + discount):5:2);
writeln('Recargo:', charge:5:2);
writeln('Total: ', (res + charge - (inscriptionP + discount)):5:2);
END.
```

7) A partir de las notas **obtenidas en dos parciales**, se determina la situación del **alumno** en la materia : **PROMOVIDO, HABILITADO o DESAPROBADO.**

Los parciales se aprueban con 4 o más.

- **Promueve el alumno que habiendo aprobado los dos parciales, obtuvo con ambos una suma de 14 y en el segundo puntaje superior a 6.**
- **Habilita el alumno que aprobó al menos un parcial con suma total mayor o igual 10, debe rendir final.**
- **Desaprueba el alumno que no cumple ninguna de las condiciones anteriores.**

Determine los datos que se requieren para establecer la condición de un alumno.

DATOS DE ENTRADA:

Set de Datos 1: Nota del Primer parcial => 8 | Nota del Segundo parcial => 8

Set de Datos 2: Nota del Primer parcial => 10 | Nota del Segundo parcial => 4

DATOS DE SALIDA:

Set de Datos 1: => El alumno **PROMUEVE**

Set de Datos 2: => El alumno esta habilitado

```
program parciales;

uses crt;

var
    // prom => Promedio
    // fParcial => Primer Parcial
    // sParcial => Segundo Parcial
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

14 Introducción a la Informática

```
prom, fParcial, sParcial: real;

BEGIN
  clrscr;
  writeln('Ingrese la nota del primer parcial');
  readln(fParcial);
  clrscr;
  writeln('Ingrese la nota del segundo parcial');
  readln(sParcial);
  clrscr;
  prom:= fParcial + sParcial;
  if ((prom >= 14) and (sParcial > 6)) then
    // Promueve el alumno al tener un promedio mayor o igual a 14 y
    su segundo parcial con una nota mayor a 6
    begin
      writeln('El alumno PROMUEVE.');
```

end

```
  else if(prom >= 10) then
    // El alumno está Habilitado para rendir finales, ya que la suma
    de ambos parciales es igual o mayor a 10
    begin
      writeln('El alumno esta HABILITADO.');
```

end

```
  else
    // Si no se cumplen cualquiera de las dos condiciones, será
    desaprobado.
    begin
      writeln('El alumno esta DESAPROBADO.');
```

end;

```
  readln();
END.
```

8) Una entidad bancaria exige una las siguiente condiciones para otorgar un crédito a un cliente.

- **Es propietario de un inmueble tasado en \$1000000 o más y tiene un sueldo menor o igual a \$1000**
- **Es propietario de un inmueble tasado en hasta \$1000000 y tiene un sueldo de más de \$1000**
- **No es propietario , pero tiene más de 5 años de antigüedad en el trabajo y gana mas de \$1500 mensuales.**

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

15 Introducción a la Informática

Determine los datos que se requieren para ser merecedor de un crédito

DATOS DE ENTRADA:

Set de Datos 1: Valor de su inmueble => 50000 | Antigüedad => 8 | Salario => 2000

Set de Datos 2: Valor de su inmueble => 1000001 | Antigüedad => 8 | Salario => 1000

DATOS DE SALIDA:

Set de Datos 1: => OTORGADO

Set de Datos 2: => OTORGADO

```
program bank;

    uses crt;

    const
        million = 1000000;

    var
        // Credit => Crédito
        credit: string;
        // OLD => Antigüedad
        old: integer;
        // Value => Valor del inmueble || Salary => Salario
        value, salary: real;

BEGIN
    clrscr;
    writeln('Ingrese el valor de su inmueble: (En caso de no tener,
    ingresar 0)');
    readln(value);
    clrscr;
    writeln('Ingrese la antigüedad de su empleo:');
    readln(old);
    clrscr;
    writeln('Ingrese su salario: ');
    readln(salary);
    clrscr;
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com


```
if((value >= million) and (salary <= 1000)) then
    // Inmueble de mayor o igual a un millón y un salario menor o
    igual a 1000
    begin
        credit:= 'OTORGADO';
    end
else if((value <= million) and (salary > 1500)) then
    // Inmueble menor o igual a un millón y salario mayor a 1500
    begin
        credit:= 'OTORGADO';
    end
else if((value = 0) and (salary > 1500) and (old > 5)) then
    // En este caso, no tiene inmueble, un salario de más de 1500 y
    una antigüedad de más de 5
    begin
        credit:= 'OTORGADO';
    end
else
    begin
        credit:= 'DENEGADO';
    end;
writeln('Salario de: $',salary:5:2);
writeln('Antigüedad: ',old);
writeln('Valor de su inmueble: $', value:5:2);
writeln('Credito: ', credit);
readln();
END.
```

9) Una empresa fumigadora, cobra a los productores distintos aranceles según el tipo de fumigación y la cantidad de hectáreas.

- Tipo 1 y 2 : 20 \$/ha.
- Tipo 3 : 30 \$/ha.
- Tipo 4 : 40 \$/ha.

Además:

Si el área a fumigar es mayor que 100 has. **se aplica un 5% de descuento.**

Si el importe **total supera los \$1.500 se aplica un 10% de descuento, sobre la cantidad que excede los \$1.500.**

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

17 Introducción a la Informática

Determine los datos que se requieren para calcular el importe a pagar por un servicio de fumigación

DATOS DE ENTRADA:

Set de Datos 1: Tipo de Fumigación => 4 | Hectáreas => 100

Set de Datos 2: Tipo de Fumigación => 1 | Hectáreas => 200

DATOS DE SALIDA:

Set de Datos 1: => 3600

Set de Datos 2: => 3400

```
program bugs;

uses crt;

var
    // sel => Selección
    sel :char;
    // selType => Tipo de Selección para informar
    selType :string;
    // res => subtotal || hec => hectáreas
    // desA => Primer descuento (Mayor a 100 hectáreas)
    // desB => Segundo descuento (Subtotal mayor a $1500)
    hec: integer;
    res, desA, desB :real;

BEGIN
    clrscr;
    writeln('Elija el tipo de fumigacion que quiere realizar: (1 - 4)');
    readln(sel);
    clrscr;
    writeln('Ingrese la cantidad de hectareas que quiere fumigar:');
    readln(hec);
    clrscr;
    if((sel = '1') or (sel = '2')) then
        begin
            res := hec * 20;
            selType := sel;
        end
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
else if(sel = '3') then
    begin
        res := hec * 30;
        selType := sel;
    end
else if(sel = '4') then
    begin
        res := hec * 40;
        selType := sel;
    end
else
    begin
        res := 0;
        selType := 'invalida';
    end;
// Primer descuento si pasa de 100 hectáreas
if (hec > 100) then
    begin
        desA := res * 0.05;
    end
else
    begin
        desA := 0;
    end;
// Segundo descuento si el subtotal pasa de 1500
if (res > 1500) then
    begin
        desB := res * 0.10;
    end
else
    begin
        desB := 0;
    end;
writeln('Selección: Tipo ', selType);
writeln('Cantidad de hectáreas: ', hec);
writeln('Subtotal: $',res:5:2);
writeln('Descuentos: $',(desA + desB):5:2);
writeln('Total: ', (res - desA - desB):5:2);
readln();
END.
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

19 Introducción a la Informática

10) Una empresa de telefonía premia con puntos a sus clientes de acuerdo al consumo registrado en el mes anterior. Los puntos permiten retirar diferentes artículos de un catálogo.

Por cada:

- **100 llamadas locales corresponden 25 puntos**
- **20 llamadas interurbanas acreditan 15 puntos**
- **5 llamadas internacionales otorgan 3 puntos**
- **Si alguna de las categorías duplica el mínimo para la obtención de puntos recibe 5 puntos más.**
- **Si acreditan puntos en las tres categorías reciben un plus de 5 puntos**

Determine **los datos que se requieren para calcular los puntos de un cliente.**

DATOS DE ENTRADA:

Set de Datos 1: Llamadas Locales => 200 | Llamadas Interurbanas => 100 | Llamadas Internacionales => 4

Set de Datos 2: Llamadas Locales => 50 | Llamadas Interurbanas => 60 | Llamadas Internacionales => 1

DATOS DE SALIDA:

Set de Datos 1: => 45 puntos.

Set de Datos 2: => 20 puntos.

```
program phonecalls;

uses crt;

const
    minLocalCalls = 100;
    minUrbanCalls = 20;
    minInterCalls = 5;

var
    // Los tres tipos de Llamadas + points => puntos
    // Categories => Cantidad de categorías acreditadas
    categories, points, localCalls, urbanCalls, interCalls: integer;
```

BEGIN

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
clrscr;
writeln('Ingrese la cantidad de llamadas locales: ');
readln(localCalls);
clrscr;
writeln('Ingrese la cantidad de llamadas interurbanas: ');
readln(urbanCalls);
clrscr;
writeln('Ingrese la cantidad de llamadas internacionales: ');
readln(interCalls);
clrscr;
points := 0;
categories := 0;
// Determina si tiene puntos por Llamadas Locales
if (localCalls > minLocalCalls) then
begin
    points:= points + 25;
    categories := categories + 1;
end;
// Determina si tiene puntos por Llamadas interurbanas
if (urbanCalls > minUrbanCalls) then
begin
    points:= points + 15;
    categories := categories + 1;
end;
// Determina si tiene puntos por Llamadas internacionales
if (interCalls > mininterCalls) then
begin
    points:= points + 3;
    categories := categories + 1;
end;

// Determina si tiene puntos adicionales por superar el doble del
mínimo por Llamadas Locales
if(localCalls > (minLocalCalls * 2)) then
begin
    points:= points + 5;
end;
// Determina si tiene puntos adicionales por superar el doble del
mínimo por Llamadas interurbanas
if(urbanCalls > (minUrbanCalls * 2)) then
begin
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

        points:= points + 5;
    end;
    // Determina si tiene puntos adicionales por superar el doble del
    mínimo por llamadas internacionales
    if(interCalls > (minInterCalls * 2)) then
        begin
            points:= points + 5;
        end;
    // Si tiene las 3 categorías acreditadas se añaden 5 puntos.
    if(categories = 3) then
        begin
            points:= points + 5;
        end;
    writeln('Cantidad de llamadas locales: ', localCalls);
    writeln('Cantidad de llamadas interurbanas: ', urbanCalls);
    writeln('Cantidad de llamadas internacionales: ', interCalls);
    writeln('Cantidad de puntos: ', points);
    readln();
END.

```

11) Un supermercado premia con descuentos las compras efectuadas por sus clientes. Por cada **\$100 descuenta \$3** y sobre el excedente que no alcanza a \$100 se aplica el siguiente descuento:

- Hasta \$20: **1.5 %**
- Más de \$20 y hasta \$50: **2%**
- Más de \$50 y menos de \$100: **2.5%**

Si la compra supera los **\$500 una vez realizados los descuentos, recibe un 1% adicional.**

Determine los datos que se requieren para calcular el importe final de una compra.

DATOS DE ENTRADA:

Set de Datos 1: Monto de la Compra => 657

Set de Datos 2: Monto de la Compra => 356

DATOS DE SALIDA:

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

Set de Datos 1: => 631.20

Set de Datos 2: => 345.60

```
program super;

uses crt;

var
    // amount => Monto de la compra
    // DiscountA => Primer descuento aplicado al cociente
    // DiscountB => Segundo descuento aplicado al módulo
    // DiscountC => Tercer descuento, aplica al subtotal -
    descuentos si el monto supera los 500
    // Quotient => Cociente
    // Remainder => Resto (Módulo)
    // Res => Total de la compra con Descuentos (Menos del descuento
    que aplica a los montos superiores a 500)
    amount, discountA, discountB, discountC, quotient, remainder,
res: real;

BEGIN
    clrscr;
    writeln('Ingrese el monto de la compra: ');
    readln(amount);
    clrscr;
    quotient := (TRUNC(amount) DIV 100);
    remainder := (TRUNC(amount) MOD 100);
    discountA := quotient * 3;
    discountB := 0;
    if (remainder <= 20) then
        begin
            discountB := 0.015 * remainder;
        end
    else if ((remainder > 20) and (remainder <= 50)) then
        begin
            discountB := 0.02 * remainder;
        end
    else if ((remainder > 50) and (remainder < 100)) then
        begin
            discountB := 0.025 * remainder;
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
end;  
res := amount - discountA - discountB;  
if (amount > 500) then  
begin  
    discountC:= res * 0.01;  
end  
else  
begin  
    discountC:= 0;  
end;  
writeln('Subtotal: $', amount:5:2);  
writeln('Descuentos: $', (discountA + discountB + discountC):5:2);  
writeln('Total: $', (res - discountC):5:2);  
readln();  
END.
```

12) La oficina meteorológica calcula la sensación térmica de acuerdo a la **temperatura, humedad, dirección (N, S, E, O) y velocidad del viento.**

- *Si la velocidad del viento es menor 10 km/h, la sensación coincide con la temperatura.*
- *Si es mayor o igual a 10 km/h y la dirección es S ó O la sensación es 3 grados menos que la temperatura, en cambio si la dirección es N ó E es 1 grado menos.*
- *Cuando el viento incide en el cálculo, si la humedad es mayor al 50% la sensación disminuye en un grado más.*

Determine los datos que se requieren para calcular la sensación térmica.

DATOS DE ENTRADA:

Set de Datos 1: Temperatura => 23 | Humedad => 99 | Velocidad del Viento => 0 | Dirección del Viento => N

Set de Datos 2: Temperatura => 10 | Humedad => 52 | Velocidad del Viento => 15 | Dirección del Viento => S

DATOS DE SALIDA:

Set de Datos 1: => 23

Set de Datos 2: => 6

program office;

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com


```
uses crt;

var
    // N, S, O, E
    windDirection: char;
    // Temperatura, humedad, velocidad del viento (wind) y sensación
    térmica
    temp, hum, wind, st: real;
    // Agregado a substraer
    stMin: integer;

BEGIN
    clrscr;
    writeln('Ingrese la temperatura actual: ');
    readln(temp);
    clrscr;
    writeln('Ingrese la humedad actual: (0-100) ');
    readln(hum);
    clrscr;
    writeln('Ingrese la velocidad del viento: ');
    readln(wind);
    clrscr;
    writeln('Ingrese la direccion del viento: (N-O-S-E) ');
    readln(windDirection);
    windDirection:= UPCASE(windDirection);
    clrscr;
    stMin:= 0;
    st:= temp;
    if((wind < 10) and (wind <> 0)) then
        begin
            if (hum > 50) then
                begin
                    stMin := stMin + 1;
                end;
            end
        else if(wind >= 10) then
            begin
                if (hum > 50) then
                    begin
                        stMin := stMin + 1;
                    end;
            end;
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

        end;
    if ((windDirection = 'S') or (windDirection = 'O')) then
        begin
            stMin := stMin + 3;
        end
    else if ((windDirection = 'N') or (windDirection = 'E'))
then
        begin
            stMin := stMin + 1;
        end;
    end;

    if (wind <> 0) then
        // En caso de que el viento no incida, no hay variación en la
        // sensación térmica
        // en función a la temperatura || temperatura = sensación
        // térmica cuando la velocidad del viento = 0
        // Y la humedad no incide no afecta a la sensación térmica
        tampoco
        begin
            st:= st - stMin;
        end;
        writeln('Temperatura: ', temp:5:2);
        writeln('Direccion del Viento: ', windDirection);
        writeln('Velocidad del Viento: ', wind:5:2);
        writeln('Humedad: ', hum:5:2);
        writeln('Sensacion Termica: ', st:5:2);
        readln();
    END.

```

13) A partir de la cantidad total de piezas que produce una máquina y la cantidad de piezas con fallas de grado 1 y grado 2 dentro de ese lote de producción.

Ejemplo: en **200** piezas hay

- **15 con fallas de grado 1**
- **5 con fallas de grado 2**

Calcular e informar:

a) Porcentaje de piezas falladas sobre el total de la producción.

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

- b) **Porcentaje de piezas con fallas de grado 1 sobre el total de piezas falladas.**
- c) **Porcentaje de piezas con fallas de grado 2 sobre el total de piezas falladas.**

Y a continuación informar la calidad de la producción de acuerdo al siguiente criterio:

- **Muy buena:** Si el porcentaje de fallas es menor al 10%.
- **Buena:** Si el porcentaje de fallas de grado 1 es menor al 20 %.
- **Regular:** Si el porcentaje de fallas de grado 1 es igual al porcentaje de fallas de grado 2.
- **Mala:** En cualquier otro caso

Determine los datos que se requieren para resolver el ejercicio.

DATOS DE ENTRADA:

Set de Datos 1: Piezas Totales => **100** | Fallas de Grado 1 => **5** | Fallas de Grado 2 => **2**

Set de Datos 2: Piezas Totales => **150** | Fallas de Grado 1 => **20** | Fallas de Grado 2 => **10**

DATOS DE SALIDA:

Set de Datos 1: => Porcentaje de Fallas => **7%** | Porcentaje de Fallas de Grado 1 => **71.43%** | Porcentaje de Fallas de Grado 2 => **28.57%** | Calidad de la producción => **Muy Buena**

Set de Datos 2: => Porcentaje de Fallas => **20%** | Porcentaje de Fallas de Grado 1 => **66.67%** | Porcentaje de Fallas de Grado 2 => **33.33%** | Calidad de la producción => **Mala**

```
program piezasM;
```

```
uses crt;
```

```
var
```

```
    // Pieces => Piezas totales
```

```
    // Fail1 => Piezas falladas de grado 1
```

```
    // Fail2 => Piezas falladas de grado 2
```

```
    // failT => Total de piezas falladas
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

    piezas, fail1, fail2, failT: integer;
    // quality => Calidad de la producción
    quality: string;
    // totalPF => Porcentaje de fallas sobre el total producción
    // pFail1 => Porcentaje de fallas de grado 1 sobre el total de
    piezas falladas
    // pFail2 => Porcentaje de fallas de grado 2 sobre el total de
    piezas falladas
    totalPF, pFail1, pFail2: real;
BEGIN
    clrscr;
    writeln('Ingrese la cantidad de piezas: ');
    readln(pieces);
    clrscr;
    writeln('Ingrese la cantidad de fallas de grado 1: ');
    readln(fail1);
    clrscr;
    writeln('Ingrese la cantidad de fallas de grado 2: ');
    readln(fail2);
    clrscr;
    // Calcular los fallos totales
    failT := fail1 + fail2;
    // Se calcula el Porcentaje de fallas sobre el total de la
    producción
    // Porcentaje total de fallas = Total de fallas * 100 / Total de las
    piezas
    totalPF := failT * 100 / piezas;
    // Se calcula el Porcentaje de fallas de grado 1 sobre el total de
    piezas falladas
    // Fallas de Grado 1 en función de las fallas totales = Total de
    fallas grado 1 * 100 / Total de fallas
    pFail1 := fail1 * 100 / failT;
    // Se calcula el Porcentaje de fallas de grado 2 sobre el total de
    piezas falladas
    // Fallas de Grado 2 en función de las fallas totales = Total de
    fallas grado 2 * 100 / Total de fallas
    pFail2 := fail2 * 100 / failT;

    if (totalPF < 10) then
        begin
            quality := 'Muy Buena';

```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
end
else if ((pFail1 < 20) and (totalPF > 10)) then
begin
    quality := 'Buena';
end
else if ((pFail1 = pFail2) and (totalPF > 10)) then
begin
    quality := 'Regular';
end
else
begin
    quality := 'Mala';
end;

writeln('Piezas totales: ', pieces);
writeln('Porcentaje de fallas: ', totalPF:4:2,'%');
writeln('Porcentaje de fallas de grado 1: ', pFail1:4:2,'%');
writeln('Porcentaje de fallas de grado 2: ', pFail2:4:2,'%');
writeln('Calidad de la produccion: ', quality);
readln();
END.
```

14) Un estacionamiento **cobra por hora \$5 los vehículos particulares y \$6 los comerciales**. Si la estadía se prolonga por más de **3 horas y hasta 12 se le efectúa un descuento del 5%** y por más de **12 se cobra una tarifa fija de \$35 y 65 respectivamente**. Si es cliente habitual (está registrado) se bonifica el precio final en 10%.

Determine los datos que se requieren para calcular el importe del estacionamiento de un vehículo.

DATOS DE ENTRADA:

Set de Datos 1: Cliente => S | Tipo de Auto => C | Estadía (Horas) => 4

Set de Datos 2: Cliente => N | Tipo de Auto => P | Estadía (Horas) => 45

DATOS DE SALIDA:

Set de Datos 1: => 20.40

Set de Datos 2: => 35.00

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
program parking;

uses crt;

const
    commercial = 6;
    particular = 5;

var
    // hours => Cantidad de horas de estadía
    hours: integer;
    // carType => Tipo de auto => su función es simplemente para
comunicarle al usuario el tipo seleccionado
    // Esta variable no es necesaria para el funcionamiento correcto
de la aplicación, pero mejora su usabilidad
    carType : string;
    // client => Tipo de Cliente || sel => Tipo de coche
    client, sel: char;
    // subtotal => Subtotal
    // discount => Descuento aplicado
    subtotal, discount: real;

BEGIN
    clrscr;
    writeln('Cliente habitual? (S - N)');
    readln(client);
    clrscr;
    writeln('Tipo de coche: (C - P)');
    readln(sel);
    clrscr;
    writeln('Cantidad de horas de la estadía: ');
    readln(hours);
    clrscr;
    //
    sel := UPCASE(sel);
    client := UPCASE(client);
    discount:= 0;
    // A partir del tipo de auto, se hace el cálculo
    case sel of
        'C':
            begin
                // Cálculo comercial
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
        carType:= 'Comercial';
        subtotal:= hours * commercial;
    end;
    'P':
    begin
        // Cálculo Particular
        carType:= 'Particular';
        subtotal:= hours * particular;
    end;
else
    begin
        // No se calcula
        subtotal:= 0;
        carType:= 'Inválido';
    end;
end;
// Horas
if ((hours <= 12) and (hours > 3)) then
    begin
        // Se calcula el descuento
        // Se podría tranquilamente utilizar el descuento cómo
        constante para "optimizar el código"
        // Pero creo funciona y nada, está perfecto así c:
        discount := subtotal * 0.05;
    end
else if (hours > 12) then
    begin
        case sel of
            'C':
                begin
                    subtotal:= 65;
                end;
            'P':
                begin
                    subtotal:= 35;
                end;
        end;
    end;
end;
// Tipo de Cliente
if (client = 'S') then
    begin
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

        // Si es cliente, se agrega el 10% (en este caso), y si hay
        un descuento anterior se agrega
        discount := discount + (subtotal * 0.10);
    end;
    writeln('Es cliente? ', client);
    writeln('Cantidad de Horas: ', hours);
    writeln('Tipo de auto: ', carType);
    writeln('Subtotal: ', subtotal:4:2);
    writeln('Descuento: ', discount:4:2);
    // Se substraee el descuento al subtotal (cálculo de horas x precio
    según el tipo de auto)
    writeln('Total: ', (subtotal - discount):4:2);
    readln();
END.

```

15) Se prueba un medicamento en un conjunto de pacientes y se registran cantidad de pacientes con resultados positivos, cantidad de pacientes con resultados negativos (empeoraron) y cantidad de resultados neutros.

Informe el resultado de la prueba de acuerdo al siguiente criterio:

- **Exitosa:** Si mejoraron más de los que empeoraron y los neutros. Y además los que empeoraron son menos que los neutros.
- **Neutra:** Si los que mejoraron son más que los que empeoraron y los neutros.
- **Fracasada:** En cualquier otro caso.

Además calcule e informe porcentaje de positivos sobre los no negativos.

Determine los datos que se requieren para resolver el ejercicio.

DATOS DE ENTRADA:

Set de Datos 1: Cantidad de Pacientes que mejoraron => 50 | Cantidad de Pacientes que no mejoraron => 15 | Cantidad de Pacientes que empeoraron => 12

Set de Datos 2: Cantidad de Pacientes que mejoraron => 25 | Cantidad de Pacientes que no mejoraron => 13 | Cantidad de Pacientes que empeoraron => 12

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

DATOS DE SALIDA:

Set de Datos 1: => Resultado de la prueba: **Exitosa** | Porcentaje de positivos sobre los no negativos: **333.33%**

Set de Datos 2: => Resultado de la prueba: **Fracasada** | Porcentaje de positivos sobre los no negativos: **192.31%**

```
program medicine;

uses crt;

var
    // success => Cantidad de Pacientes que mejoraron
    // neutral => Cantidad de Pacientes que no mejoraron
    // failures => Cantidad de Pacientes que empeoraron
    success, neutral, failures: integer;
    // Porcentaje
    percentage :real;
    // qualification => Resultado del Experimento
    qualification :string;
BEGIN
    clrscr;
    writeln('Ingrese la cantidad de pacientes que mejoraron: ');
    readln(success);
    clrscr;
    writeln('Ingrese la cantidad de pacientes que no mejoraron: ');
    readln(neutral);
    clrscr;
    writeln('Ingrese la cantidad de pacientes que empeoraron: ');
    readln(failures);
    clrscr;
    // Indicar el resultado del experimento
    if ( (success > (neutral + failures)) and (neutral > failures))
then
    begin
        qualification := 'Exitosa';
    end
    else if (success > (neutral + failures)) then
    begin
        qualification := 'Neutra';
    end
    else
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

begin
    qualification := 'Fracasada';
end;
// Calcular el porcentaje de positivos sobre los no negativos
// Teniendo en cuenta que Positivos => success | no negativos =>
neutral | negativos => failures
if (success <> 0) then
    begin
        percentage := success * 100 / neutral;
    end
else
    begin
        percentage := 0;
    end;

    writeln('Resultado de la prueba: ', qualification);
    writeln('Porcentaje de positivos sobre los no negativos: ',
percentage:4:2, '%');
    readln();
END.

```

16) Una empresa de transportes realiza envíos de cargas, **se desea calcular y mostrar el importe a pagar.**

Los datos que se ingresan son **PESO** de la carga (número real) y **CATEGORÍA** (dato codificado: **1 - común, 2 - especial, 3 - aéreo**).

El precio se calcula a **\$2 por Kg para categoría común, \$2.5 por Kg para categoría especial y \$3 por Kg para categoría aérea.**

Se cobra recargo por sobrepeso: 30% si sobrepasa los 15 Kg, 20% si pesa más de 10 Kg y hasta 15 Kg inclusive, 10% más de 5 Kg y hasta 10 Kg inclusive.

DATOS DE ENTRADA:

Set de Datos 1: Peso => 10 | Categoría => 2

Set de Datos 2: Peso => 20 | Categoría => 3

DATOS DE SALIDA:

Set de Datos 1: => 30

Set de Datos 2: => 78

program transport;

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```

uses crt;
const
    // Valores por categoría
    common = 2;
    special = 2.5;
    aerial = 3;
    // Sobrecargos por peso
    // Si sobrepasa Los 15KG
    charge3 = 0.30;
    // Si sobrepasa entre más de 10KG y hasta 15KG
    charge2 = 0.20;
    // Si sobrepasa entre más de 5KG y hasta 10KG
    charge1 = 0.10;

var
    // Weight => Peso | Subtotal => Monto a pagar | addCharge =>
    Sobrecargo por peso
    weight, subtotal, addCharge: real;
    // category => Categoría
    category :char;
    // Variable adicional, solo funcional a la usabilidad del
    programa
    selCategory :string;
BEGIN
    clrscr;
    writeln('Ingrese el peso del paquete: ');
    readln(weight);
    clrscr;
    writeln('Ingrese el metodo de envio: ');
    writeln('1 - Comun ');
    writeln('2 - Especial ');
    writeln('3 - Aereo ');
    readln(category);
    clrscr;
    // Se calcula el precio sin sobrecargos
    case category of
        '1':
            begin
                selCategory := 'Comun';
                subtotal := weight * common;
            end;
    end;

```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com

```
'2':
    begin
        selCategory := 'Especial';
        subtotal := weight * special;
    end;
'3':
    begin
        selCategory := 'Aerea';
        subtotal := weight * aerial;
    end;
else
    begin
        selCategory := 'Invalida';
        subtotal := 0;
    end;
end;
// Se calcula si hay sobrecargos
if (weight < 5) then
    begin
        addCharge:= 0;
    end
else if ((weight >= 5) and (weight < 10)) then
    begin
        addCharge:= subtotal * charge1;
    end
else if ((weight >= 10) and (weight < 15)) then
    begin
        addCharge:= subtotal * charge2;
    end
else
    begin
        addCharge:= subtotal * charge3;
    end;
writeln('Peso de la carga: ', weight:4:2);
writeln('Metodo de Envio: ', selCategory);
writeln('Subtotal: $', subtotal:5:2);
writeln('Recargo por peso: $', addCharge:5:2);
writeln('Total: $', (subtotal + addCharge):5:2);
readln();
END.
```

NAKANISHI RICARDO ANDRÉS

LICENCIATURA EN GESTIÓN DE SISTEMAS Y NEGOCIOS

+54 223 300 0563

andresnakanishi@gmail.com