

Las siguientes consignas han sido recopiladas de finales de mesas anteriores.

### **Resolución de problemas, dato y tipos de datos**

1. Se lee una secuencia de 10 números enteros. Escribí un programa que informe la cantidad de dígitos mayores, menores e iguales a 4 del total de números, y para cada número la suma de todos sus dígitos. Además que informe el número con mayor cantidad de dígitos.

En este ejemplo tratá de detallar las etapas que se debieron cumplir para llegar a escribir el programa.

Etapla 1: Leer el enunciado y abstraer el problema.

Etapla 2: Generar un modelo.

Etapla 3: Separar el problema en subproblemas.

Etapla 4: Escribir el código.

---

```

program Ejercitacion1;
procedure descomponer(n : integer; var cant,total: integer);
begin
    cant := 0;
    total := 0;
    while (n > 0) do
    begin
        total := total + n mod 10;
        cant := cant + 1;
        n := n div 10;
    end;
end;
var
    i,iguales,menores,mayores,max,numeroMax,n,
    sumaDeDigitos,cantidadDeDigitos : integer;

begin
    max := -1;
    iguales:=0;menores :=0; mayores:=0;
    for i:= 1 to 10 do
    begin
        read(n);
        if (n < 4) then
            menores := menores + 1
        else
            if (n = 4) then
                iguales := iguales + 1
            else
                mayores := mayores + 1;
            descomponer(n,cantidadDeDigitos,sumaDeDigitos);
            write(sumaDeDigitos);
            if (cantidadDeDigitos > max) then
            begin
                max := cantidadDeDigitos;
                numeroMax := n;
            end;
        end;
        write(menores,iguales,mayores,numeroMax);
    end.

```

2. Se lee una sucesión de números enteros terminados con el 9999, se desea obtener:

- a) La suma de todas las unidades.
- b) La suma de los tres últimos dígitos de cada número.
- c) El promedio de los números leídos.

Ejemplo: se leen los números 132, 4201, 372, 23025, 9999 (no se procesa)

La respuesta será a) 10, b) 6-3-12-7, c) 5546.

```

program ej2;
function ultimosTres(n : integer):integer;
var
    tot,i : integer;
begin
    tot := 0;
    i := 0;
    repeat
        tot := tot + n mod 10;
        n := n div 10;
        i := i + 1
    until (i = 3) or (n = 0);
    ultimosTres := tot;
end;

var
    cont, n, suma : integer;
begin
    suma := 0;
    cont := 0;
    read(n);
    while (n <> 9999) do
        begin
            suma := suma + n;
            write (ultimosTres(n));
            cont := cont + 1;
            read(n);
        end;
    write (suma,suma/cont);
end.

```

3. Escribí un programa que lea una secuencia de caracteres terminada en punto e informe cuántas palabras contienen exactamente tres vocales distintas.

Además informá cuáles son las vocales de cada palabra.

(Puede estar mal)

```

program ej3;
var
  c,i : char;
  vocalesEncontradas, vocales : set of char;
  palabras,cantVocales: integer;
begin
  palabras := 0;
  vocales := ['A','E','I','O','U','a','e','i','o','u'];
  repeat read(c) until (c <> ' ');
  while (c <> '.') do
    begin
      cantVocales := 0;
      vocalesEncontradas := [];
      while (c <> ' ') and (c <> '.') do
        begin
          if (c in vocales) then
            begin
              if not (c in vocalesEncontradas) then
                begin
                  cantVocales := cantVocales + 1;
                  vocalesEncontradas := vocalesEncontradas + [c];
                end;
            end;
          read(c);
        end;
      for i := 'a' to 'z' do
        if (i in vocalesEncontradas) then
          write('Se encontró ', i);
        if (cantVocales = 3) then palabras := palabras + 1;
      repeat read(c) until (c <> ' ');
    end;
    write ('Palabras con 3 vocales: ',palabras);
  end.

```

4. Se tiene una secuencia de palabras, terminadas en punto. Cada palabra está separada por un único blanco. Se lee además una palabra de teclado. Se quiere saber cuántas palabras de la secuencia son **parecidas** a la palabra leída. Donde **parecidas** significa que coincidan en letra y posición en 3 ó más caracteres.

Ejemplo: "solero sol soleado sal soltero asolado solo."

la palabra leída: sol

Cantidad de palabras parecidas a sol: 5.

(tampoco funciona)

```

program ej4;
var
  texto, leida : string;
  i,j : integer;
  tam,cont, palabrasParecidas : integer;
begin
  texto := 'solero sol soleado sal soltero asolado solo.';
  readln (leida);
  palabrasParecidas := 0;
  tam := length(leida);
  i := 1;
  while (texto[i] <> '.') do
    begin
      cont := 0;
      j := 1;
      while (texto[i] <> '.') and (texto[i] <> ' ') and (j < tam) do
        begin
          if (texto[i] = leida[j]) then cont := cont + 1;
          i := i + 1;
          j := j + 1;
        end;
      if (cont >= 3) then palabrasParecidas := palabrasParecidas + 1;
      if (j = tam) then
        while (texto[i] <> '.') and (texto[i] <> ' ') do
          i := i + 1;
        end;
      writeln(palabrasParecidas);
    end.
end.

```

5. Dado el siguiente programa: informar que imprime en cada caso.

**program ejercicio;**

**var** alfa,beta , gama, epsilon: integer;

**procedure** calcular (alfa: integer; var gama : integer; var beta: integer; var epsilon: integer)

**begin**

alfa:= beta +1;

beta:= alfa + 5;

gama:= beta + 15;

epsilon:= beta - gama;

write (alfa); write (beta); write (gama); write (epsilon);

**end;**

**begin**

alfa:= 13; beta:= 16; gama:= -6; epsilon:= 4;

calcular(epsilon,alfa,beta,gama);

write (alfa); write (beta); write (gama); write (epsilon);

**end.**

**Solución:**

Dentro del proceso al principio:

alfa = 4, gama = 13, beta = 16, epsilon = -6.

luego alfa = 17, beta = 9, gama = 24, epsilon = -13.

Estos últimos valores son los que se imprimirán DENTRO DEL MÓDULO

alfa, beta y gama tomarán los valores gama, beta, epsilon del módulo respectivamente según su posición, dado que son utilizados por referencia mientras que los cambios en epsilon no serán reflejados en el programa principal

Entonces, alfa imprimirá 24, beta imprimirá 9, gama imprimirá -13 y epsilon 4.