

Práctico 2 - Entrada/Salida

Programación 1

InCo - Facultad de Ingeniería, Udelar

1. Considere la siguiente secuencia de instrucciones y que todas las variables se han declarado como variables enteras:

```
read(x,y,z);  
readln(a);  
readln(b,c);  
read(d);
```

Suponga que se ejecuta dicha secuencia con los siguientes datos ingresados por la entrada estándar:

```
8 7 2 1 3  
1 4 4 6 2  
3 7
```

Indique con qué valores quedan las variables a, b, c, d, x, y, z.

a=1, b=1, c=4, d=3, x=8, y=7, z=2

2. Dada la siguiente declaración de variables:

```
var  
  a,b,c: integer;  
  x,y,z: real;
```

Suponga que se ingresan los siguientes datos por la entrada estándar:

```
3 2.3 -6.5  
1 5 2.1
```

Para cada una de las siguientes secuencias de instrucciones, indique si se ejecutan correctamente o si se produce error en tiempo de ejecución. En caso afirmativo, indique en qué instrucción se produce el error.

- I) `read(x,y,z); read(a,b,x)`
- II) `read(a,x,y); read(b,z,c)`
- III) `readln(a,x); read(b,c,y,z)`

Todas se ejecutan sin error. La última no termina y queda esperando más datos en la entrada para leer el valor de z

3. ¿Qué se desplegará en la salida estándar cuando se ejecute el siguiente programa en Pascal? Anticipe cuál va a ser la salida que se mostrará. Después, ejecútelo en máquina y compare lo que esperaba con la salida que se muestra.

```
program Valores_suma;
begin
  write ('Valor1 es: ');
  writeln(3);
  write ('Valor2 es: ');
  writeln(5.5);
  writeln ('La suma de ', 3, ' y ', 5.5, ' es ', 3 + 5.5)
end.
```

Compile y ejecute el programa para obtener la solución.

4. Suponga que a, b y c son variables enteras y se ingresan los siguientes datos por la entrada estándar:

```
1 2
3 4
```

Indique cuál o cuáles de las siguientes secuencias de instrucciones **no** producirá los valores $a = 1$, $b = 2$ y $c = 3$:

- | | |
|--|---|
| <input type="checkbox"/> read(a); readln(b); read(c) | <input type="checkbox"/> read(a); read(b); readln; read(c) |
| <input type="checkbox"/> readln (a,b,c) | |
| <input type="checkbox"/> read(a,b,c) | <input checked="" type="checkbox"/> readln(a); readln(b); readln(c) |

5. Dado el siguiente programa:

```
program Char_Ord;
var a, b, c, d : char;
    r1, r2 : integer;
    resu : real;
begin
  read (a, b, c, d);
  r1 := ord(a) - ord('0');
  r2 := (ord(c) - ord('0')) * 10 + (ord(d) - ord('0'));
  resu := r1 + r2 / 100;
  writeln ('Valor ingresado: ', resu:4:2)
end.
```

Suponga que se ejecuta dicho programa con los siguientes datos ingresados por la entrada estándar:

3.14

¿Qué se desplegará en la salida estándar cuando se ejecute el programa? Anticipe cuál va a ser la salida. Después, ejecútelo en máquina y compare lo que esperaba con la salida que se muestra.

Notar que para anticipar la salida, no es necesario conocer el valor que retorna la función *ord* para cierto caracter. Basta con saber que los caracteres '0', '1', ..., '9' están en posiciones consecutivas en la tabla ASCII. Entonces, por ejemplo, sabemos que $\text{ord}('1') - \text{ord}('0') = 1$.

Compile y ejecute el programa para obtener la solución.

6. Escriba un programa en Pascal que, dados dos enteros de la forma hhmm (ingresados por la entrada estándar), donde hh representa las horas (menos de 24) y mm los minutos (menos de 60), determine la suma de estos dos tiempos, y exhiba el resultado en la salida estándar, en la forma d hhmm, donde d es días (ya sea cero o uno). Incluya mensajes de salida con etiquetas descriptivas para el ingreso de datos.

Ejemplo:

Entrada	Salida
1345 2153	1 1138

```
program Pr2Ej6;
var
  (* hora y minutos de cada hora ingresada *)
  h1, m1, h2, m2 : Integer;
  hora1, hora2 : Integer;
  dias, horas, minutos : Integer;
begin
  Writeln('Ingrese las horas a sumar');
  Readln(hora1,hora2);

  (* Separacion de horas y minutos *)
  h1:= hora1 div 100;
  m1 := hora1 mod 100;
  h2:= hora2 div 100;
  m2 := hora2 mod 100;

  minutos := (m1+m2) mod 60;

  (* Suma de horas *)
  horas:= (h1+h2 + (m1+m2) div 60) mod 24;
```

```

(* El valor de dias sera 0 o 1,
   dependiendo de si la suma de horas supera 24 *)
dias := (h1+h2+(m1+m2)div 60) div 24;

Writeln(dias:1, ' ', horas:2,minutos:2)
end.

```

7. El error relativo en una medición m es la razón de la diferencia absoluta entre la medición y el valor verdadero v al valor verdadero, la cual se calcula como $|m - v|/v$. Escriba un programa en Pascal que lea por la entrada estándar dos números reales que representan la medición m y el valor verdadero v . El programa debe calcular el error relativo de la medición y exhibirlo en la salida estándar junto con la medición y el valor verdadero. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplo:

Entrada	Salida
51.0 51.3	Medición = 5.1000000000e+01 Valor verdadero = 5.1300000000e+01 Error relativo = 5.8479532164e-0003

```

program Pr2ej7;
var
  e,m,v : real;
begin

  Writeln('Ingrese valores de tipo real de medición y valor verdadero');
  Readln(m,v);

  (* error relativo *)
  e := abs(m-v) / v;

  Writeln('Medicion es = ', m);
  Writeln('Valor verdadero es = ', v);
  Writeln('Error relativo es = ', e)
end.

```

8. No es posible utilizar una computadora para generar números aleatorios genuinos ya que es preciso utilizar un algoritmo para generar los números, lo que implica que es posible predecir los números generados. Lo que sí pueden hacer las computadoras es generar números pseudoaleatorios (números que, estadísticamente, parecen ser aleatorios). Una técnica antigua (que no produce buenos resultados) se llama método del cuadrado medio. Funciona así: dado un número a , para generar el siguiente número de la secuencia se extraen los dígitos que están en la posición de las decenas y las centenas de a^2 . Por ejemplo, si a es

53, entonces a^2 es 2809, y el siguiente número pseudoaleatorio será 80. Se ve que el siguiente número pseudoaleatorio a 80 es 40. Si se continúa este proceso se obtiene 60, 60, 60, ...

Escriba un programa en Pascal que lea por la entrada estándar un entero de dos dígitos y determine el siguiente número pseudoaleatorio que se generaría si se usara el método del cuadrado medio. Como resultado, exhiba por la salida estándar el número de dos dígitos original, el cuadrado de este entero, y el siguiente número pseudoaleatorio. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplo:

Entrada	Salida
53	Numero introducido = 53 Cuadrado del numero = 2809 Siguiete numero pseudoaleatorio = 80

```

program Pr2Ej8;
var
  num, numcuadrado, signum : integer;
begin
  Writeln('Ingrese número');
  Readln(num);

  numcuadrado := sqr(num);

  signum := numcuadrado div 10;

  signum := signum mod 100;

  Writeln('Numero introducido = ', num);
  Writeln('Cuadrado del numero = ', numcuadrado);
  Writeln('Siguiete numero pseudoaleatorio = ', signum)
END.

```

9. La función `exp` de Pascal calcula un valor igual a la suma de la serie infinita $1 + [x/1!] + [(x^2)/2!] + [(x^3)/3!] + [(x^4)/4!]$, siendo x un número real. Escriba un programa en Pascal que lea por la entrada estándar un valor real para x entre 0,0 y 1,0. El programa debe determinar, por un lado, la suma de los primeros cinco términos de la serie infinita y, por otro lado, el valor de `exp(x)` mediante la función predefinida. Como resultado, exhiba por la salida estándar ambos valores calculados junto con el valor de x . Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplo:

Entrada	Salida
0.5	Valor introducido: x = 5.0000000000e-01 Suma de los cinco términos = 1.648437619e+00 Valor de Exp (x) = 1.6487212707e+00

```

program Pr2Ej9;
var
  num,suma : real;
begin

  Writeln('Ingrese número real en intervalo [0.0,1.0]');
  Readln(num);

  (* suma de los cinco primeros términos de la serie *)
  suma := 1 + num + sqr(num)/2 + sqr(num)*num/(2*3) +
                                                sqr(sqr(num))/(2*3*4);

  Writeln('Numero introducido = ',num);
  Writeln('Suma de la serie= ',suma);
  Writeln('Exp(x)= ', exp(num))
END.

```

10. Con el resultado del ejercicio 11 del Práctico 1, escriba un programa en Pascal para determinar la raíz cuadrada de un número real positivo a mediante el cálculo de $a^{0.5}$. El programa deberá leer el valor para a por la entrada estándar. Como resultado, exhiba por la salida estándar el valor de a , el valor de $a^{0.5}$, y el valor de $\text{sqr}(a)$ (el cual se calculará con la función predefinida). Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplo:

Entrada	Salida
12.7	Valor introducido: a = 1.2700000000e+01 Raiz cuadrada calculada = 3.5637059362e+00 Valor de Sqrt (a) = 3.5637059362e+00

```

program Pr2Ej10;
var
  num,raiz : real;
begin

```

```

Writeln('Ingrese número real positivo');
Readln(num);

(* Se aplica solución definida en el Ejercicio 11 del Práctico 1 *)
raiz := exp(0.5*ln(num));

Writeln('Numero introducido = ', num);
Writeln('Raiz cuadrada calculada= ', raiz);
Writeln('Sqrt(x)= ', sqrt(num))
END.

```

11. Escriba un programa en Pascal que transforme un número octal (base ocho) de tres dígitos en su correspondiente número equivalente en base 10. Por ejemplo, el número octal de tres dígitos 415 equivale al valor decimal $4 \times 8^2 + 1 \times 8^1 + 5 \times 8^0$, o sea, 269. El programa deberá leer por la entrada estándar el número octal y realizar las operaciones aritméticas necesarias para calcular el número equivalente en base 10. Como resultado, exhiba por la salida estándar el valor octal ingresado junto con el número equivalente en base 10. Incluya mensajes de salida con etiquetas descriptivas para el ingreso y el despliegue de datos.

Ejemplo:

Entrada	Salida
217	Octal 217 = Decimal 143

```

program Pr2Ej11;
const base = 8;
var
  octal, decimal, d1, d2, d3 : Integer;
begin
  Writeln('Ingrese un numero en base ', base:1, ' de 3 digitos: ');
  Readln(octal);

  (* Separacion de dígitos *)
  d1 := octal div 100;
  d2 := (octal div 10) mod 10;
  d3 := octal mod 10;

  (* Evaluación del polinomio *)
  decimal := d1*sqr(base) + d2*base + d3;

  Writeln('Octal ', octal:3, ' = Decimal ', decimal:3)
END.

```