

Segundo Parcial. Programación 1

Instituto de Computación

noviembre 2006

Observaciones:

- Todos los programas o fragmentos de programas, deben ser escritos en el lenguaje **Pascal** tal como fue dado en el curso. A grandes rasgos este es el Pascal estándar con algunos agregados a saber:
- Utilización de `else` en la instrucción `case`. Si no se especifica `else` se ejecutará la sentencia siguiente al `case` en caso de no coincidir con ninguna de las etiquetas.
- Evaluación por circuito corto de los operadores booleanos AND y OR.
- En todos los problemas se evaluará además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera se restarán puntos entre otros conceptos por: mala o nula indentación, mala utilización de las estructuras de control, código confuso e innecesariamente largo, programas ineficientes, utilización de variables globales, pasaje incorrecto de parámetros, etc. No obstante, por razones prácticas no exigimos que incluya comentarios en los códigos que escriba en la prueba.
- Escriba su nombre completo y cédula en todas las hojas. Numere todas las hojas entregadas y escriba en todas la cantidad total de hojas.
- Escriba de un solo lado de cada hoja, un ejercicio por hoja.

Ejercicio 1

Dados dos números enteros A mayor que 0 y B mayor que 1, existe un único par de enteros e y r tales que:

- I. $B^e \times r = A$
- II. r no es divisible por B .

Hacer un programa que ingrese A y B desde la entrada y despliegue los valores de e y r.

Se puede asumir que los números A y B cumplen con las condiciones requeridas, no es necesario validarlos. No se pueden invocar ninguna de las funciones aritméticas predefinidas del pascal (exp, log, sqr, sqrt, etc.).

Ejemplos:

A	B	e	r
1000	5	3	8
500	5	3	4
1000	6	0	1000
100	200	0	100
100	25	1	4

Ejercicio 2

Escribir un programa que lea un texto de la entrada y determine cuál es la secuencia de caracteres consecutivos iguales más larga. El programa desplegará el carácter y la cantidad de veces que este se repite. El texto se ingresa en una sola línea y termina con un punto que no debe ser considerado como parte del texto (centinela). Si existe más de una secuencia máxima se considera la que aparece primero.

Puede asumirse que el texto siempre tiene al menos un carácter además del punto.

Ejemplos:

Entrada:
aaabbaccabbbba.

Salida:
Carácter: b
Veces: 4

Entrada:
bcaaabbaccabba.

Salida:
Carácter: a
Veces: 3

Ejercicio 3

Sea un arreglo definido de la siguiente manera:

```
type ArregloNum = array [1..N] of integer;
```

Donde N es una constante entera mayor que 1.

Escribir una función

```
function EntreNegativos(a: ArregloNum): integer
```

que retorna el índice i del primer elemento de a que cumpla:

- $a[i] \geq 0$
- $a[i-1] < 0$
- $a[i+1] < 0$

Si no existe tal elemento, retorna N+1.

La función no debe recorrer celdas innecesariamente.

Ejercicio 4

Se consideran los siguientes tipos:

```
const M = {algun valor > 1}
type
  TipoMes = (enero, febrero, marzo, abril,
            mayo, junio, julio, agosto,
            setiembre, octubre,
            noviembre, diciembre);
  TipoFecha = record
    dia: 1..31;
    mes: TipoMes;
    anio: 0..3000;
  end;
  ArregloFechas = array [1..M] of TipoFecha;
```

Se pide escribir un procedimiento:

```
procedure HallarMinMax(fechas: ArregloFechas;
                      var min,max: TipoFecha);
```

que retorna en los parámetros min y max las fechas más vieja y más nueva respectivamente que aparecen en el parámetro fechas. El procedimiento debe recorrer una sola vez el arreglo.

Ejercicio 5

Determinar la salida del siguiente programa, cuando se le da como entrada el dígito de las unidades (el último dígito antes del guión) de su número de cédula. Por ejemplo, si su cédula es 1234567-8, la entrada será 7.

```
PROGRAM segundoParcial;
VAR z:INTEGER;
Procedure primerAuxiliar(VAR a:Integer;VAR b:Integer);
VAR z : integer;
BEGIN
    z:= a+1;
    a:= z;
    b:= b*2;
END; (*primerAuxiliar*)

Procedure segundoAuxiliar(a:Integer; b:Integer);
BEGIN
    z:= a+1;
    b:= b*2;
END; (*segundoAuxiliar*)

Procedure tercerAuxiliar(a:Integer;VAR b:Integer);
BEGIN
    a:= a+1;
    b:= b*2;
    WriteLn(a,' ',b,' ',z);
    segundoAuxiliar(a,b);
END; (*tercerAuxiliar*)

BEGIN
    readln(z);
    tercerAuxiliar(z,z);
    WriteLn(z);
    primerAuxiliar(z,z);
    WriteLn(z);
END.
```