

Bucles

Fundamentos de la programación y la informática

Grado en ingeniería aeroespacial en vehículos aeroespaciales 2019-2020

Escuela Técnica Superior de Ingeniería de Telecomunicación

Universidad Rey Juan Carlos

Práctica 7.1. Bucle while 1

Escribe un programa en un fichero llamado `~/fpi/practica07/mientras01.pas` que, usando la sentencia *while*, vaya generando números reales aleatorios entre 0 (incluido) y 1 (excluido), y mostrándolos en pantalla. Este proceso se repetirá mientras el número obtenido sea estrictamente menor que una constante (local) llamada *Objetivo*. Dale por ejemplo el valor 0.9. Al finalizar, el programa indicará el número de números que ha sido necesario generar.

Práctica 7.2. Bucle while 2

Escribe un programa en un fichero llamado `~/fpi/practica07/mientras02.pas` que, usando *while*, vaya lanzando dados y sumando el total de puntos acumulados. El programa seguirá lanzando y sumando mientras la suma de puntos sea inferior a la constante *Objetivo*. El número de caras del dado será la constante *CarasDado*. Ambas constantes serán locales al cuerpo del programa principal, dales el valor que prefieras.

Práctica 7.3. Lectura de teclado

Escribe un programa en el fichero `~/fpi/practica07/hasta01.pas` que pida al usuario un número entero positivo, las veces necesarias hasta que el usuario realmente escriba un entero positivo. Usa *val* y *repeat*.

Práctica 7.4. Lecturas de teclado

Escribe un programa que muestre los mensajes *Escribe un número real entre 0 y 1 (o escribe 'f' para finalizar)*, *Escribe otro número real entre 0 y 1 (o escribe 'f' para finalizar)*.

El programa hará lo esperado: tras cada petición, leerá un número, las veces necesarias hasta que el usuario cumpla las instrucciones. Tras leer ambos, mostrará el producto de los números. Todo esto se repetirá las veces necesarias hasta que el usuario escriba 'f' en algún momento. (Es tradicional acabar este tipo de programas con la *f* de *fin* o la *q* de *quit*).

Práctica 7.5. Aleatorios negativos

Escribe un programa en un fichero llamado `~/fpi/practica07/negativos.pas` que genere números aleatorios enteros entre $-N$ y $+N$ (ambos inclusive), donde N será una constante local al cuerpo del programa principal. Este programa estará basado en una función que:

- Generará un número entero aleatorio entre 0 y N .
- Multiplicará el número anterior por -1 en la mitad de las ocasiones. Esto se puede hacer por ejemplo generando enteros aleatorios entre 1 y 2, ambos inclusive. O generando reales entre 0 y 1 y comprobando si el valor es menor a 0.5.

El programa generará un total de M números (constante local) y estará basado en *for*.

Práctica 7.6. Números aleatorios dentro de intervalo

Escribe un programa en un fichero llamado `~/fpi/practica07/intervalo_reales.pas` que genere números reales aleatorios dentro del intervalo (a, b) (intervalo cerrado por la izquierda y abierto por la derecha).

Sobra decir que tienes que escribir una función que genere el número, que el cuerpo del programa principal llamará a la función las veces necesarias y mostrará el resultado.

Observa que puedes conseguir esto:

1. Partiendo de a
2. Sumándole un valor aleatorio que será, como mínimo, 0, y como máximo, la diferencia entre a y b .
En otras palabras, $random() * (b - a)$

Ejemplo: $a = 10$, $b = 15$. Necesitamos números aleatorios entre 0 y $(15 - 10 = 5)$, que sumaremos a 10. Esto es, multiplicar 5 por un número aleatorio entre 0 y 1.

Observa que si alguno de los dos extremos es negativo, esto sigue funcionando.

El programa generará un total de M números (constante local) y estará basado en *for*.

Práctica 7.7 Números enteros dentro de intervalo

Escribe un programa en un fichero llamado `~/fpi/practica07/intervalo_enteros.pas` que genere números enteros aleatorios dentro del intervalo (a, b) **ambos incluidos**. Haz este programa copiando el anterior y retocándolo ligeramente. Observa que los intervalos que proporciona *random* son abiertos por la derecha y aquí lo necesitamos cerrado. Pista: tendrás que sumar 1 *en alguna parte*.

El programa generará un total de M números (constante local) y estará basado en *for*

Práctica 7.8. Máximo y Mínimo desde teclado

Escribe un programa en un fichero llamado `~/fpi/practica07/maximo_minimo.pas` que vaya pidiendo números reales positivos al usuario. Cuando se introduzca un 0, el programa debe terminar indicando antes el máximo y el mínimo de los números introducidos.

Si el usuario introduce algo diferente a lo solicitado (la entrada no es un número o no es un número real positivo) el programa debe ignorarlo, escribir un mensaje de error y seguir funcionando con normalidad, solicitando un nuevo número.

Si el usuario no introduce ningún número positivo antes del 0, el programa terminará pero no debe mostrar el máximo ni el mínimo, sino un mensaje de error específico.

Ejemplo de funcionamiento:

```
./maximo_minimo
Escribe un número positivo (0 para terminar): jj
Valor no válido.
Escribe un número positivo (0 para terminar): 12.5
Escribe un número positivo (0 para terminar): -1
Valor no válido.
Escribe un número positivo (0 para terminar): 8.0
Escribe un número positivo (0 para terminar): 23.9
Escribe un número positivo (0 para terminar): 0
Máximo: 23.9
Mínimo: 8.0
```

Observaciones

- Tendrás que repetir un *procedimiento principal* hasta que el usuario indique el final.
- En ese procedimiento principal, tendrás que repetir la lectura de un valor hasta recibir una entrada *válida*, donde *válido* significa o bien un 0, o bien un número real positivo. Naturalmente, esto será un procedimiento.

Práctica 7.9. Esperanza matemática de un dado

Escribe un programa en un fichero llamado `~/fpi/practica07/esperanza.pas` que calcule la esperanza matemática de un dado, a partir de su número de caras. Este número será una constante en el cuerpo del programa principal. Ejemplo: La esperanza matemática de un dado de 6 caras es $\frac{1+2+3+4+5+6}{6} = 3,5$