**GUIA DE ESTUDIO**

| MATERIA: Algorítmica I | CURSO: | Segundo |
| --- | --- | --- |
| FECHA: 19/08/2024 | PROFESOR: | Ing.Carmen Keim |

**CAPACIDADES**

Conoce la lógica del funcionamiento de los números aleatorios Implementa la función random en ejercicios propuestos en Dev-C++

Conoce la lógica de la implementación de menús y los aplica en ejercicios propuestos

**TEMA**

**Números aleatorios y Menús ACTIVIDADES**

¡Muy buenas tardes colegas!

Esta tarde vamos a dar lectura primeramente al material que les dejo adjunto.

**Estoy pendiente a sus dudas y consultas**

**¡Les deseo un Buen Trabajo! EJERCICIOS**

Numeros Aleatorios

1. Simular el lanzamiento de una moneda al aire e imprimir si ha salido cara o cruz. Repetir el proceso tantas veces se desee.
2. Simular cien tiradas de un dado y contar las veces que aparece el número 6.
3. Simular cien tiradas de dos dados y contar las veces que entre los dos suman 10.
4. Simular el lanzamiento de una moneda al aire y que pare cuando salgan 3 caras seguidas. Imprimir el número de tiradas realizadas.
5. Simular la carrera de dos caballos, si cada uno tiene igual posibilidad de ganar (**considerar la función gotoxy, es exclusivo para investigar)**

Menus:

1. Introducir dos números por teclado y mediante un menú calcular su suma, resta, producto o división. Ejemplo:

Menú de Opciones:

* 1. Suma
  2. Resta
  3. Multiplicación
  4. División
  5. Salir del Programa Elige una opción:

1. Introducir un número que represente un mes del año. Imprimir el mes al que corresponde

utilizando la instrucción CASE.

1. Ingresar un numero entre 20 y 50. A partir de esto, seleccionar los elementos del siguiente menú y realiza las opciones indicadas:
   1. Serie del 1 al número ingresado
   2. Serie del número ingresado al 100
   3. Serie del número ingresado al 1
   4. Salir del programa Elige una opción:
2. Seleccionar los elementos del siguiente menú y realiza las opciones según se indica a continuación:
   1. **Figuras**
      1. Cuadrado

Ingresar el lado de un cuadrado y calcular mediante las opciones:

* + - 1. Perímetro
      2. Área
    1. Rectángulo

Ingresar el lado de un cuadrado y calcular mediante las opciones:

* + - 1. Perímetro
      2. Área
      3. Regresar al menú anterior
  1. Series
     1. Creciente

Ingresar un numero entre 20 y 30, a partir de este valor mostrar la serie de acuerdo a lo indicado en el menú:

* + - 1. Serie del 1 al número ingresado
      2. Serie del número ingresado al 100
      3. Decreciente

Ingresar un número entre 20 y 30, a partir de este valor mostrar la serie de acuerdo a lo indicado en el menú:

1. Serie del número ingresado al 1
2. Serie del 100 al número ingresado
3. Regresar al menú anterior
   1. Aleatorio
      1. Simular el lanzamiento de una moneda y decir cuál de las caras ha caído
      2. Simular el lanzamiento de un dado. Imprimir el número que ha caído
      3. Regresar al menú anterior
      4. Salir del Programa

# Algorítmica III Números Aleatorios

En determinadas ocasiones no es necesario introducir números desde el exterior, podemos indicarle al ordenador que los genere el interiormente. Son lo que se llama números aleatorios.

* Los números aleatorios son números creados internamente por el ordenador.
* El valor de los números está comprendido entre cero y la unidad.
* La forma de generarlos depende de cada lenguaje de programación.

***Sintaxis:***

# RND

El numero generado por RND es mayor que “0” y menor que “1”.

Para generar números enteros mayores a la unidad y comprendidos entre unos límites, lo expresaremos de la siguiente forma:

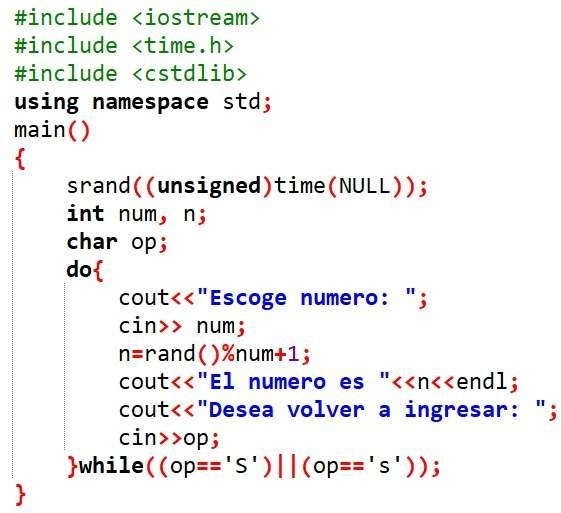
Int(RND\*numero) + 1

Int(RND\*(Limite Superior – Límite inferior) ) + Límite Inferior

* + Int: parte entera
  + RND: genera los números aleatorios
  + Numero: valor máximo de los números a generar, empezando desde el cero (Pero en nuestra fórmula uno le sumamos 1, lo que hará que empiece desde 1)
  + Se le suma la unidad para generar desde el 1 al número máximo deseado
  + Límite superior e inferior: se utilizan en caso de que se quiera generar números aleatorios entre un rango de datos. Ejemplo: entre 20 y 50, ya que la primera fórmula genera los números aleatorios a partir del 1.

# En Dev-C++

Ejemplo del ingreso de un número “n” para calcular números aleatorios desde el 1 al número ingresado, el cual se repite hasta que el usuario lo decida.



Consideraciones importantes:

* La biblioteca “Time.h” es necesario ya que esto nos permite tomar un valor arbitrario desde el motor de Dev-C++ que trabaja con el tiempo y esto combinado con la función rand nos permite obtener números totalmente aleatorios
* La biblioteca “cstdlib” permite utilizar la instrucción “unsigned” el cual nos permite tomar solo valores positivos

# Menús

- Con la instrucción **srand((unsigned) time(NULL));** nos permite declarar la función random indicándole que solo nos interesa los valores positivos y que los números aleatorios estarán vinculados a la función Time del motor de Dev-C++

Las operaciones a realizar por el momento se están preparando por separado sin ningún nexo de unión entre ellas. Generalmente esto no sucede así; es preciso realizar una serie de operaciones que pueden estar más o menos relacionadas entre sí, indicando cual es la que queremos ejecutar en ese instante.

Un menú consiste en presentar en pantalla una ventana con una serie de operaciones u opciones a realizar, cada una de las cuales realiza una función determinada.

Cuando termine de ejecutar cada una de ellas, ya sea en la función principal o subprogramas, el programa vuelve de nuevo al menú del que había partido.

A veces los menus se presentan anidados, es decir, alguna de las opciones del menú al ser seleccionada hace que aparezca otro menú, dando lugar a nuevas posibilidades de selección.

Los menus permiten ejecutar mas de un programa , sin necesidad de tener que escribir su nombre, cada vez que se desea ejecutarlo. Simplemente le indicamos mediante una variable la opción deseada.

La selección del programa a realizar se puede hacer mediante la instrucción IF, pero hay una forma más rápida y fácil de manejar, el cual es a través la instrucción condicional múltiple CASE.

Sintaxis:

# En dev C++

Seleccionar CASO(numero) CASO 1

Realizar acción 1

CASO 2

Realizar acción 2

CASO 3

Realizar acción 3 CASO n

Realizar acción n CASO contrario

Realizar acción alternativa en caso de que ninguna se haya cumplido Fin seleccionar

