



Estructuras de datos: Arrays unidimensionales

Vectores: Proyecto colaborativo

Notación algorítmica.

Definición de tipo

tipo
array[L:U] **de** <tipo_dato>: <identificador_tipo_array>

Declaración de tipo

<identificador_tipo_array>: lista_de_variables

Proyecto: Hágalo Ud. Mismo.

1. Especificación Pseudocódigo.

Diseñe los siguientes algoritmos respectivos, bajo los siguientes requerimientos:

- Defina y declare vectores como contenedores de datos.
- Implemente operaciones de recorrido para el tratamiento de datos.
- Implemente la generación números aleatorios para simular fuentes de datos externos
- Defina e implemente sub-rutinas para aquellas tareas que considere importantes.

2. Especificación UML.

Diseñe los Activity Diagramas (Diagramas UML de actividades) para los respectivos algoritmos.

3. Comprobación de Algoritmos.

Implemente el modo proyecto para la comprobación de los algoritmos en código C++.

ENUNCIADOS.

Ejemplo 7. Números aleatorios. Modificar en el ejemplo 5, el algoritmo `tempMedia_01`: el bloque de acciones bajo el comentario: `//bucle entrada de datos`

- Sustituir **el bucle entrada de datos** por un procedimiento de generación al azar de temperaturas en el rango de 21.00 a 24.00 para el mes de Septiembre.

NOTA: Ver documento Practica 01 Estructuras de Datosy Arrays Uni

Ejemplo 8. Números aleatorios. Use y extienda el ejemplo 6, el algoritmo `Lanzar_dados_al_azar`, y aplíquelo en juegos de Dados, como el conocido **Siete**, por ejemplo.

Ejemplo 9. Encontrar el valor máximo, mínimo y promedio de una serie de 100 números reales generados en forma aleatoria en el rango: 20.00 a 60.00 y almacenados en un array tipo vector.

Análisis.

Para resolver este problema, necesitamos bucles de repetición, para lectura de la fuente de datos, y los cálculos estadísticos solicitados:

- Implementar una operación de lectura de un array tipo vector (Unidimensional) en donde se asignen mediante números aleatorios los 100 valores reales en el rango 20-60.
- Lleno el vector con valores, implementar 3 operaciones de recorrido del vector para calcular: El promedio, el máximo, y el mínimo de la serie.

Especificaciones de E/S.

Entradas: 100 números aleatorios leídos y almacenados en un vector de tipo real

Salidas : máximo, mínimo y promedio

OBSERVACION: Tome como modelo de ayuda el siguiente algoritmo (diseño parcial) y luego complete el algoritmo aplicando el método modular.

1.a. Pseudocódigo. Diseño parcial del algoritmo.

```
//Driver: programa principal
Algoritmo Estadística de reales
tipo
    array[1..100] de real: arrSerie
var
    //Declaraciones

inicio
    //invocar sub-rutinas
    llamar_a leerArray(serie, n)           //sub-rutina tipo procedimiento
    media ← promedio(serie, n)             //sub-rutina tipo funcion
    llamar_a maximoMinimo(serie, n, max, min) //tipo procedimiento
    llamar_a visualizarDatos(media, max, min) //tipo procedimiento
fin

//Módulos: definir prototipos de sub-rutinas a continuacion.
```