Algoritmos y Estructuras de Datos II
Tipos de datos

Dr. Edgard Iván Benítez Guerrero
cursofei@gmail.com

Tipo de dato

- Definición del conjunto de valores que puede tomar una variable (dominio)
- Especificados
  - En extensión: {1,2,3}
  - En intención
    - Entero
    - Flotante
    - Cadena de caracteres
    - Arreglos
    - Estructuras (registros)
- Restricción impuesta para la interpretación, manipulación y representación de datos

Tipos de datos abstactos (TDA)

- Colección de datos y un conjunto de operaciones permitidas (acciones) que son usadas para definir y manipular datos
- Un TDA combina la abstracción de los datos y los programas. Un TDA se especifica por:
  - Un nombre del tipo único
  - Un conjunto de valores
  - Un conjunto de operaciones que pueden actuar sobre el tipo de dato
- Ejemplo: entero

Tipos en lenguajes de programación

- La asignación de tipos a los datos tiene dos objetivos principales:
  - detectar errores en las operaciones
  - determinar cómo ejecutar estas operaciones
- Clases de lenguajes de programación según la tipificación:
  - Lenguaje fuertemente tipificado: todos los datos deben tener un tipo declarado.
  - Lenguajes no tipificados: no se requiere que los datos tengan un tipo declarado explícitamente.

Clasificación de tipos de datos

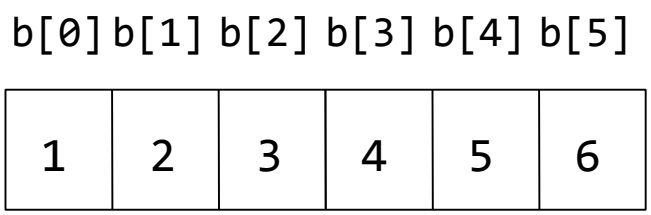
- Estáticos: el tamaño que ocupa una variable en memoria no puede cambiar en tiempo de ejecución
  - Simple: tipos básicos ofrecidos en general por los lenguajes de programación
    - Ordinales: el conjunto de valores que representa se puede contar (entero, lógico, carácter)
    - No ordinales: el conjunto de valores que representa no se puede contar (reales)
  - Estructurado:
    - Colecciones ordenadas (arreglos) y no ordenadas (conjuntos) de elementos de un mismo tipo
    - Estructuras conteniendo elementos de tipos diferentes
- Dinámicos: el tamaño de que ocupa una variable en memoria puede ser modificado en tiempo de ejecución (apuntadores)

Repaso de tipos de datos elementales en C

Tipo	Ejemplo	Tamaño en bytes	Rango mínimo..máximo
char	‘c’	1	0..255
short	-15	2	-128..127
int	1024	2	-32768..32767
unsigned int	42325	2	0..65535
long	262144	4	-2147483648.. -2147483647
float	10.5	4	$3.4*(10^{-38})$.. $3.4*(10^{38})$
double	0.00045	8	$1.7*(10^{-308})$.. $1.7*(10^{308})$
long double	1e-8	8	Igual que double

Arreglos

- Secuencia de objetos del mismo tipo
- Declaración en C:
  - <tipo> <nombre>[<tamaño>]
- Los datos almacenados pueden ser de cualquier tipo que se pueda definir en C
- Ejemplos:
  - int c[5] = {1,2,3,4,5}; // Se inicializa
  - int b[] = {1,2,3,4,5,6}; // Se puede omitir su tamaño



Arreglos para representar cadenas de caracteres

- En C no existe el tipo « cadena »: se usan arreglos de tipo char, donde el caracter '\0' indica el fin de cadena
- Ejemplos:
  - char c1[20] = {'H','o','l','a',0};
  - char c2[20] = "Hola cadena";

Arreglos de dos o más dimensiones

- Un arreglo 2D es conocido como matriz:
  - `int m1[2][3] = {{1,2,3},{2,2,1}};`
  - `float m3[][4] = {{0.2, 1.1, 2.3, 3.1},`
    `{1.1, 1.2, 1.3, 5.4}};`

- Es posible tener arreglos en 3 o más dimensiones
  - `int espacio[2][2][2];`
  - `int penta[2][3][2][3][4];`

Estructuras

- Colección de uno o más elementos denominados miembros, cada uno de los cuales puede ser de un tipo diferente
- Ejemplo:

```
typedef struct {
    char nombre[30];
    int edad;
} Persona;
Persona p;
```

- La notación "." se usa para acceder a cada campo de la estructura

```
p.nombre = "Francisco Mina";
p.edad = 24;
```