

Clase #04 de 27

Expresiones Aritméticas & Iteración

Constantes Simbólicas

Abril 03, Lunes

Repaso Clase Anterior

- Análisis de hello.c
- Autómata para Identificadores
- Glosario
- Tareas

Agenda para esta clase

- Estado de Equipos
- Expresiones & Iteraciones
- Sintaxis & Semántica de for y while
- Constantes simbólicas
- Tipos de Datos
- Funciones



Diego Esquivel
diegoEsquivel16

Follow

Block or report user

Overview

Repositories 6

Stars 0

Followers 0

Following 0

Popular repositories

Esquivel-Calculator

Calculator Despagar.com

Java

FinalProyect-Esquivel

Minesweeper

Java

dientito

Ruby

toy_app

Ruby

sample_app

Ruby

HelloWorldC

TP N° 0 SSL - Diego Esquivel

C

Expresiones & Iteraciones

K&R 1.2-1.3 Variables, Tipo de datos, Expresiones Aritméticas y la Sentencia For

Problema – Tabla Fahrenheit-Celsius

$$^{\circ}\text{C} = 5/9 (^{\circ}\text{F} - 32)$$

0	-17
20	-6
40	4
60	15
80	26
100	37
120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

Resolución – Fahrenheit-Celsius

$$^{\circ}\text{C} = 5/9 (^{\circ}\text{F} - 32)$$

```
/* F2C K&R 1988 */
#include <stdio.h>
int main(void){
    int fahr, celsius;
    int lower, upper, step;

    lower = 0;    /*scale lower limit*/
    upper = 300;  /*upper limit*/
    step = 20;    /*step size*/
    fahr = lower;

    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr,
celsius);
        fahr = fahr + step;
    }
}
```

- Comentarios
- Variables
 - Abstracción de datos
 - Tipo de datos
 - Declaraciones de variables
 - Anuncio de a propiedades
 - Asociación
 - Múltiples declaraciones
- Comienzo de la Ejecución (Cómputo)
- Sentencias
- "Sentencia de asignación"
- Sentencia while
- Sentencia compuesta
- Indentación
 - Estilos de codificación
- Secuencia
- Expresiones aritméticas
- División entera y real
- Formateo de la salida
 - ¿Cómo mejorar la alineación?.

F-C 2 – Formato y Precisión

```
#include <stdio.h>
int main(void){
    float fahr, celsius;
    int lower, upper, step;

    lower = 0;    /* lower limit of temperature scale */
    upper = 300; /* upper limit */
    step = 20;    /* step size */
    fahr = lower;

    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```


Otros formatos para printf

- Diferencia entre Valor y Formato
 - 65 , 41, 101, LXV, A
 - 1000001
- %d entero decimal
- %6d entero decimal, por lo menos ancho 6
- %f punto flotante
- %6f punto flotante, por lo menos ancho 6
- %.2f punto flotante y 2 caracteres luego del punto
- %6.2f punto flotante, por lo menos ancho 6 y 2 caracteres luego del punto
- %o octal
- %x hexadecimal
- %c carácter
- %s string
- %% por ciento

Ejercicios

- 1-3. Encabezado sobre la tabla
- 1-4. C-F.

Sintaxis & Semántica de for y while

F-C 3 – For generaliza While y una "Best Practice"

```
#include <stdio.h>
int main(void){
    float fahr, celsius;
    int lower, upper, step;

    lower = 0;
    upper = 300;
    step = 20;
    fahr = lower;

    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```



```
#include <stdio.h>

int main(void){
    int fahr;

    for( fahr = 0; fahr <= 300; fahr = fahr + 20 )
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```

For versus While

`/* Una sentencia for equivalente a una sentencia while*/`

`while (expresión) sentencia`

`for (; expresión ;) sentencia`

`/* "Una" sentencia while equivalente a una for */`

`for (expresión1; expresión2 ; expresión3) sentencia`

`expresión1;`

`while (expresión2) { sentencia expresión3; }`

Sintaxis y Semántica Simple de While y de For

- While
 - Sintaxis
 - $Sentencia_{While} \rightarrow \text{while} (Expresión) Sentencia$
 - $Sentencia \rightarrow \dots$
 - $Expresión \rightarrow \dots$
 - Semántica
 - $\text{while} (Expresión_1) Sentencia_1$
 - LN
- For (C90)
 - Sintaxis
 - $Sentencia_{For} \rightarrow \text{for} (Expresión_{opt} ; Expresión_{opt} ; Expresión_{opt}) Sentencia$
 - Semántica
 - $\text{for} (Expresión_{1_{opt}} ; Expresión_{2_{opt}} ; Expresión_{3_{opt}}) Sentencia_1$
 - LN.

Sintaxis y Semántica Simple de While y de For

- For (C90)
 - Sintaxis
 - $SentenciaFor \rightarrow \text{for} (Expresión_{opt} ; Expresión_{opt} ; Expresión_{opt}) Sentencia$
 - Semántica
 - $\text{for} (Expresión1_{opt} ; Expresión2_{opt} ; Expresión3_{opt}) Sentencia1$
 - LN
- For (C99)
 - Sintaxis
 - $SentenciaFor \rightarrow \text{for} (Expresión_{opt} ; Expresión_{opt} ; Expresión_{opt}) Sentencia$
 - $SentenciaFor \rightarrow \text{for} (Declaración1 Expresión_{opt} ; Expresión_{opt}) Sentencia$
 - Semántica
 - $\text{for} (Expresión1_{opt} ; Expresión2_{opt} ; Expresión3_{opt}) Sentencia1$
 - LN.
 - $\text{for} (Declaración1 Expresión_{opt} ; Expresión_{opt}) Sentencia$
 - LN.

Intervalo

20 minutos

Constantes simbólicas

K&R 1.4 Constantes Simbólicas

“Números mágicos”

```
#include <stdio.h>
```

```
int main(void){  
    int fahr;
```

```
    for( fahr = 0; fahr <= 300; fahr = fahr + 20 )  
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
```

```
    return 0;  
}
```

Constantes (o Nombres) Simbólicas

```
#include <stdio.h>
#define LOWER 0    /* lower limit of table */
#define UPPER 300  /* upper limit */
#define STEP 20    /* step size */

int main(void){
    int fahr;

    for( fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP )
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```

C11: Calificador const y Declaración en for

```
#include <stdio.h>
int main(void){
    const int LOWER = 0;    /* lower limit of table */
    const int UPPER = 300; /* upper limit */
    const int STEP  = 20;   /* step size */

    for(int fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```

- Ejercicio 1-5. Modifique el programa para que imprima la tabla en orden inverso, es decir, desde 300 hasta 0 grados.

Funciones del Preprocesador, Parte I

- Incluir archivos encabezado, directiva `#include`
- Reemplazar comentarios por un espacio
- Atender directivas `#define` que definen nombres macro y su lista de reemplazo
- Expandir los nombres macro
- Concatenar cadenas adyacentes.

Tipo de dato



Algunos tipos de datos

- Tipos Enteros, ordenados por rango
 - **bool** Lógico (C99, stdbool.h)
 - **char** Código
 - **int** Palabra
 - **short** Corto
 - **long** Largo
- Tipos Flotantes Reales, ordenados por rango
 - **float** Simple precisión
 - **double** Doble precisión
- El tamaño depende de la máquina, aunque hay restricciones y mínimos
- Variantes de **int**
 - **unsigned**
 - **short**
 - **long**
 - **long long** (C99)
- Variantes de **double**
 - **long**
- Tipos Flotantes complejos
 - **float _Complex**
 - **double**
- **_Complex**
- **long double _Complex**
- Tipos Aritméticos
 - Tipos Enteros
 - Tipos Flotantes, Reales y Complejos
- Tipos Básicos
 - **char**
 - Tipos enteros
 - Tipos flotantes
- Enumeraciones
 - **enum**
- Tipos Derivados (de los anteriores)
 - Arreglos
 - Estructuras
 - Uniones
 - Punteros a todos ellos.

Funciones

K&R 1.7-1.8 Funciones y Argumentos

Funciones

- Programa
 - Conjunto / Secuencia de funciones
- Abstracción procedural, inclusive para funciones de una línea
 - Qué versus Cómo
- Recomendables y Eficientes, inclusive cortas
 - Aplicar inclusive para funciones de una línea
- Prototipo de función
- Definición de función, en cualquier orden y en diferentes fuentes
- Valor y tipo retornado
- C versus Pascal.

Funciones (cont.)

```
#include <stdio.h>
int power(int m, int n);
/* test power function */
int main(void){
    int i;

    for( i=0; i<10; ++i )
        printf(
            "%d %d %d\n",
            i,
            power(2,i),
            power(-3,i)
        );
}
```

- Prueba con programa simple (Driver Program)
- Prototipos y sus parámetros
- Argumentos son expresiones
- Invocaciones son expresiones
- Ignorar valor de la invocación
- Argumento versus parámetro

```
/* power: raise base
   to n-th power;
   n >= 0 */
```

```
int power(int base,int n){
    int i, p;

    p = 1;

    for( i=1 ; i<=n ; ++i )
        p = p * base;

    return p;
}
```

- Definición de función
- Precondición y Poscondición, Dominio e Imagen
- Localidad de parámetros
- **return** *expresión* ;
- Control de flujo.

Pasaje por valor (o copia)

```
/* power: raise base
   to n-th power;
   n >= 0; version 2 */

int power(int base, int n){
    int p;

    for( p=1 ; n>0 ; --n )
        p = p * base;

    return p;
}
```

- No es desventaja, es ventaja
- ¿Solo in?
- Parámetros son variables
- Efecto de lado acotado
- Punteros para simular el pasaje por referencia
- Pasaje de arreglos.

Ejercicio

- 1-15. Reescriba el programa de conversión de temperatura de la sección 1.2 para que use una función de conversión.

Términos de la clase #04

Definir cada término con la bibliografía

- Expresiones & Iteraciones
 - Abstracción de datos
 - Declaraciones
 - División entera y real
 - Operación cerrada
 - Valor de una expresión
 - Efecto de lado de una expresión
 - Formateo de valor
- Sintaxis & Semántica de for y while
 - For generaliza a While
 - While particulariza a For
 - For versus While
 - Especificación de Sintaxis
 - Gramática
 - Especificación de Semántica
 - Sintaxis y Semántica de Sentencia for
 - Sintaxis y Semántica de Sentencia while
 - For C99
- Constantes simbólicas
 - “Números mágicos”
 - Constantes simbólicas ó Nombres simbólicos
 - Directiva #define
 - Calificador const
- Tipos de Datos
 - Tipos Enteros
 - Tipos Flotantes
 - Variantes de tipo
 - Tipos Aritméticos
 - Tipos "Básicos"
 - Tipos Aritméticos
 - Tipos Derivados
 - Arreglos
 - Estructuras
 - Uniones
 - Punteros
 - Precisión: Moneda
- Funciones
 - Abstracción procedural
 - Función
 - Prototipo o Declaración de función
- Expresión de Invocación
 - Argumento
 - Definición de Función
 - Parámetro
 - Nomenclatura (estilo)
 - Precondición
 - Dominio
 - Poscondición
 - Imagen
 - Programa de prueba (Driver Program)
 - Pasaje por valor (copia)
 - Pasaje por variable (referencia)
 - Efecto de lado de una función
 - Localidad de parámetros
 - Sentencia de salto return
 - Simulación de pasaje por referencia (variable).

Tareas para la próxima clase

1. Comenzar Trabajo #1: Conversor de Temperatura

¿Consultas?

Fin de la clase