Clase #05 de 27

Expresiones & Iteraciones, Constantes Simbólicas, y Fases de la Traducción

Abril 22, Martes
Abril 23, Miércoles

Agenda para esta clase

- Expresiones & Iteraciones
- El Preprocesador
- Constantes Simbólicas
- Trabajo #1 "Fases de la Traducción y Errores"

Expresiones & Iteraciones

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

Problema – Tabla Fahrenheit-Celsius

$°C = 5/9 (°F - 32)$

```
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

$$°C = 5/9 (°F - 32)$$

```
// F2C K&R 1988
#include <stdio.h>
int main(){
 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 propiedades
  - Asociación
  - Declaración con varios Declaradores
- 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(){
  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

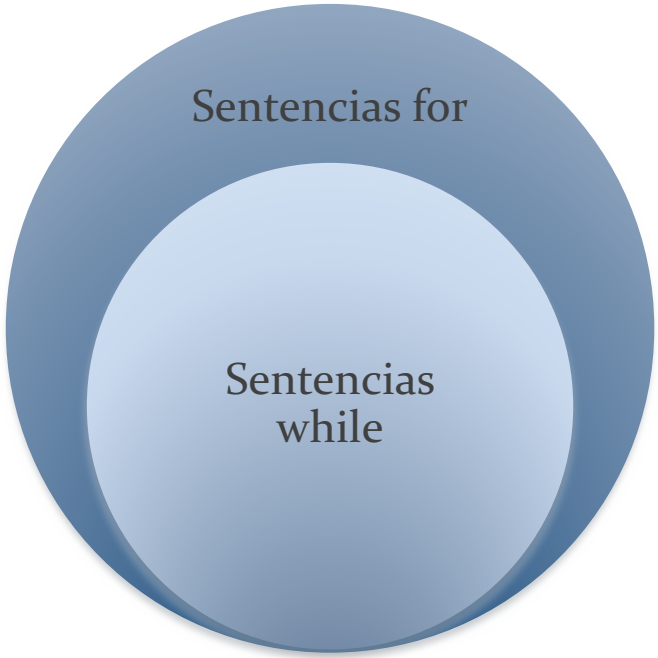
- Diferencia entre Valor y Formato
  - 65 , 41, 101, LXV, A
  - 1000001
- %d entero decimal
- %i también entero decimal, la diferencia está en la función scanf
- %6d entero decimal, por lo menos ancho 6
- %f punto flotante
- %6f punto flotante, por lo menos ancho 6
- %.2f punto flotante y 2 caracteres 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
- %% porciento

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;
  }
}
```



- for por while
- Variable celsius reemplazada por expresión

```
#include <stdio.h>

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

Ejercicios

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

El Preprocesador

Dos Funciones Básicas del Preprocesador

```
#include <_____.h>

return/*entre*/0;
return0;
return 0;
```

- Incluir archivos, directiva `#include`
- Reemplazar comentarios.

Preprocesador: Función “polémica”

- `include`: “copy & paste”
- `define`: “Buscar & Reemplazar”
  - Atender directivas #define que definen nombres macro y su lista de reemplazo
  - Expandir los nombres macro
    - `#define MAX 10`
    - `#define MAX(A,B) a > b ? a : b`
- Best Practice
  - Evitar, en los posible, el uso define, buscar alternativas más abstractas y con misma eficiencia.
    - http://www.stroustrup.com/bs_faq2.html#macro
    - Alternativas
      - `enum`
      - `const`
      - `constexpr`
      - `inline`

Constantes simbólicas

K&R 1.4 Constantes Simbólicas

“Números mágicos”

```
#include <stdio.h>

int main(){
  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(){
  int fahr;

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

Calificador const (C90) y Declaración en for (C99)

```
int main(){
  const int LOWER = 0,   // lower limit of table
            UPPER = 300, // upper limit
            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));
}
```

Calificador constexpr (C23)

```
int main(){
  constexpr int LOWER = 0,   // lower limit of table
                UPPER = 300, // upper limit
                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.

Fases de la Traducción y Errores

Trabajo #1

Trabajo #1

7

Fases de la Traducción y Errores

7.1. Objetivos

Este trabajo tiene como objetivo identificar las fases del proceso de traducción o *Build* y los posibles errores asociados a cada fase.

Para lograr esa identificación se ejecutan las fases de traducción una a una, se detectan y corrigen errores, y se registran las conclusiones en `readme.md`.

No es un trabajo de desarrollo; es más, el programa que usamos como ejemplo es simple, similar a `hello.c` pero con errores que se deben corregir. La complejidad está en la identificación y comprensión de las etapas y sus productos.

7.2. Temas

- Fases de traducción.
- Preprocesamiento.
- Compilación.
- Ensamblado.
- Vinculación (Link).
- Errores en cada fase.

Términos de la clase #05

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
- Preprocesador
  - #include
  - Tratamiento de comentarios por parte del preprocesador
- Constantes simbólicas
  - “Números mágicos”
  - Constantes simbólicas ó Nombres simbólicos
  - Directiva #define
  - Definición y Expansión
  - Calificador const
  - Calificador constexpr
  - Best Practice: Evitar defines
  - Alternativas a #define
  - `enum`
  - `const`
  - `constexpr`
  - `inline`
- Fases de la Traducción y Errores

Tareas para la próxima clase

1. Leer hasta **1.4 Constantes Simbólicas** inclusive.
2. Comenzar con el trabajo #1 "Fases de la Traducción y Errores", para entregar en dos semanas.

¿Consultas?

Fin de la clase