



Informática Aplicada I

Marcela Morales Quispe

Septiembre 25, 2013

Características de un programa

- Integridad
- Claridad
- Simplicidad
- Eficiencia
- Modularidad-
Reusabilidad
- Generalidad

Palabras reservadas

Estas palabras tienen un significado especial para el compilador de C, por lo que el programador debe de tener cuidado de no utilizar estas palabras como nombre de variables o identificadores

- auto
- break
- case
- char
- const
- continue
- default
- do
- double
- else
- enum
- extern
- float
- for
- goto
- if
- int
- long
- register
- return
- short
- signed
- sizeof
- static
- struct
- switch
- typedef
- union
- unsigned
- void
- volatile
- while

Estructura de selección múltiple

Su objetivo es comprobar varios valores constantes posibles para una expresión.

```
switch (expression)
{
    case constant1:
        group of statements 1;
        break;
    case constant2:
        group of statements 2;
        break;
    .....
    default:
        default group of statements;
}
```

- Si algún caso coincide, se inicia la ejecución de ese caso.
- Todos los casos deben de ser diferentes.
- La sentencia **default** es opcional, si no se encuentra y si ninguno de los casos coincide simplemente no se hace nada.
- La sentencia break indica al programa una salida inmediata de la estructura switch. Después de ejecutarse algún caso el programa continúa a menos que se indique explícitamente abandonar el flujo.

Sentencias *break* y *continue*

- La sentencia ***break*** se utiliza para modificar el flujo de control, cuando se ejecuta en una estructura ***for***, ***while*** y ***do-while*** o ***switch***, causa la salida inmediata de dicha estructura. La ejecución del programa continúa con el primer enunciado después de la estructura.
- Los usos comunes de la sentencia ***break*** son para escapar en forma prematura de un ciclo, o para saltar el resto de una estructura ***switch***.
- La sentencia ***continue*** cuando se ejecuta en una estructura ***for***, ***while*** y ***do-while***, salta los enunciados restantes del cuerpo de dicha estructura.
- (Ver ejemplos).

Operadores Lógicos

exp1	exp2	exp1 && exp2
0	0	0
0	1	0
1	0	0
1	1	1

Tabla de verdad para el operador && (AND lógico)

exp1	exp2	exp1 exp2
0	0	0
0	1	1
1	0	1
1	1	1

Tabla de verdad para el operador || (OR lógico)

expresión	!expresión
0	1
1	0

Tabla de verdad para el operador ! (negación lógica)

Funciones

- Las funciones permiten al programador modularizar un programa.
- Todas las variables declaradas en las definiciones de función son variables locales. La mayor parte de estas funciones tienen una lista de parámetros, éstos proporcionan una forma de comunicación entre funciones.

```
Tipo-de-valor-de-regreso nombre-de-la-función(lista de parámetros)
{
    Declaraciones;
    Sentencia1;
    Sentencia2;
    Sentencian;

    return valor;
}
```


- ***return*** sirve para retornar un valor (o el valor de una expresión) cuando llamamos a una función.
- ***return expresión***, se suele usar (***expresión***) pero es opcional.
- Un prototipo de función le indica al compilador el tipo de dato regresado por la función, el número de parámetros que la función espera recibir, los tipos de dichos parámetros y el orden en el cual se esperan dicho parámetros.

Ejemplo

```
#include<stdio.h>
int maximo(int, int);
int main()
{
    int a,b,c;
    printf("Ingresa dos números:");
    scanf("%d %d ", &a, &b);
    printf("El valor máximo es:%d", maximo(a,b));
    return 0;
}
int maximo(int x, int y)
{
    int max;
    if(x > y)
        max = x;
    else
        max = y;

    return max;
}
```