Programación

Ejercicios Tema 3
Elementos Básicos del Lenguaje C

Autores:

M. Paz Sesmero Lorente
Paula de Toledo Heras
Fco. Javier Ordoñez Morales
Juan Gómez Romero
Jose A. Iglesias Martínez
Jose Luis Mira

SOLUCIONES

Segunda sesión: precedencia de operadores, cadenas, punteros

1. **Escriba un programa que pida una temperatura en grados Farenheit y la pase a Celsius**.

Solución:

```
#include <stdio.h>
int main(void){
   float celsius,farenheit;
   //Se solicita y se lee la temperatura en Farenheit
   printf("\nDeme la temperatura en grados Farenheit\n");
   scanf("%f",&farenheit);
   //Se calcula la temperatura en celsius: (farenheit -32)*5/9
   celsius=(farenheit-32)*5/9;
   //El resultado obtenido se muestra por pantalla.
   printf("\nLa temperatura en grados celsius es: %f\n",celsius);
   return (0);
}
```

2. **Escriba un programa que muestre en la pantalla un mensaje de presentación, pregunte al usuario su nombre y le salude con un mensaje personalizado que use el nombre que se acaba de leer**.

Solución:

```
#include <stdio.h>
int main(void){
  char nombre[256];
  printf ("Buenos dias, Como se llama usted?\n");
  scanf ("%s",nombre); /*Las cadenas NO van precedidas de &*/
  printf ("\nHola %s\n",nombre);
  return (0);
}
```

3. **Escriba un programa en el que se declare una variable de tipo entero y se le asigne un valor. El programa debe mostrar el valor de la variable y la dirección de memoria en la que se almacena**.

Solución:

```
#include <stdio.h>
int main(void){
  int num=6;
  //Se define un puntero a una variable de tipo entero:
  int *puntero;

  printf ("\nLa variable tiene como valor:%i\n",num);
   /*Se almacena en puntero la dirección de memoria en la que se
   almacena la variable num.*/
  puntero=&num;
   /*Se muestra por pantalla el valor almacenado en puntero y el
   valor almacenado en num. Para el puntero se usa el descriptor de
   formato %p, específico para punteros*/
  printf ("\nLa direccion de memoria donde se encuentra num
es:%p\n", puntero);
  return (0);
}
```

4. ¿Cuál es el resultado de evaluar la siguiente expresión? Suponga que la variable n tiene el valor 6.0 y la variable valor 2.0 y ambos son de tipo float. Escriba la expresión equivalente a esta utilizando paréntesis.

```
minut = 25.0 + 120 * n / valor
```

Solución:

a) Si minut es de tipo float su valor es 385.000
b) Si minut es una variable de inpo int su valor es 385
c) Utilizando paréntesis, la expresión sería

```
minut = 25.0 + ((120 * n) / valor)
```

5. Encuentre los errores del siguiente programa

```
int main(void)
{
    foat radio, perimetro;
    printf ( introduzca el radio");
    scanf("%f", &radio);
    perimetro= 2*PI*radio
    printf("%f", perimetro);
    return(0)
```

Solución:

- Falta una llave de cierre en la función main.
- Falta un ; en la instrucción `return(0)`.
- foat por float.
- Faltan las comillas en la cadena de caracteres "introduzca el radio"

6. Escriba un programa en C que asigne a la variable x (de tipo entero) el resultado de 4/0. ¿Qué ocurre al compilar el programa? ¿Y al ejecutarlo?

Solución:

```
#include <stdio.h>

int main(void){

  int x;
  x=4/0;

  return (0);
}
```

- Al compilar el programa se genera el siguiente mensaje en la línea 6:
  [Warning] division by zero
- Al intentar ejecutarlo se genera una excepción.

7. Escriba un programa en C que lea dos enteros en las variables x e y, y calcule y muestre por pantalla los valores de x/y (como número real) y de x%y
Ejecute el programa introduciendo valores diferentes. ¿Qué sucede cuando a la variable y se le asigna el valor 0?

Solución:

```c
#include <stdio.h>

int main(void){
  int x, y;
  float cociente;
  int resto;

  printf ("Introduzca los valores de x e y ");
  scanf ("%d %d", &x, &y);
      /*cociente=x/y;  División entre enteros. El resultado es
entero.
      Al asignar este valor a una variable real la parte decimal se
completa    con 0*/
  cociente=(float)x/(y);  //Division entre reales. Resultado real
  printf ("El cociente es: %.2f\n", cociente);
      /*El especificador de formato %.2f hace que el cociente se
escriba con       dos decimales*/
  resto=x%y;
  printf ("El resto es: %d\n", resto);

  return (0);
}
```

Al asignar a la variable **y** el valor **0** se genera una excepción (error en tiempo de ejecución).

8. Escriba un programa que lea tres enteros (a, b, c) y muestre por pantalla un 1 si los valores introducidos siguen un orden creciente (a>b>c) y 0 en caso contrario.

Solución:

```c
/*Objetivo: Mostrar el uso de los operadores lógicos y demostrar
que a>b>c no equivale a ((a>b)&&(b>c))*/
#include <stdio.h>

int main(void)
{
   int a, b, c;
   printf ("Introduzca 3 números enteros\n");
   scanf("%d %d %d", &a, &b, &c);
   printf ("Si los valores introducidos siguen un orden creciente
se mostrara un 1\n");
   printf("%d\n", ((a<b)&&(b<c)));
  /* La sentencia (a>b>c) calcula a>b y compara el resultado de
   esta operación con c).*/

   return (0);
}
```

9. Escriba un programa que declare una variable de tipo entero x y una variable y de tipo real, asigne a dichas variables los valores 6 y 2.0, respectivamente, y calcule y muestre por pantalla el resultado de las siguientes operaciones:

a) x*y
b) x/y
c) x%y

¿Qué sucede al intentar compilar el programa? ¿Cómo resolvería este problema?

Solución:

```
#include <stdio.h>
int main(void)
{
  int x;
  float y;
  //Se asignan a x e y los valores indicados.
  x=6;
  y=2.0;

  //Se transcribe el código indicado.
  x*y;
  x/y;
   /* x%y; Al poner esta sentencia se genera un error de
   compilación: Invalid operands to binary %. El operador % sólo
   funciona con operandos de tipo entero. Para subsanar este error
   convertimos la variable y a tipo entero:*/
  x%(int)y;

  return (0);
}
```

10. Escriba y compile el siguiente programa en C:

```
#include <stdio.h>

int main(void){
  int a, b;
  char cadena[8];
  int c;
  a=7; b=14; c=128;
  printf ("Asigne un valor a la cadena de caracteres ");
  scanf("%[^\n]", cadena); /*El descriptor %[^\n] lee los
caracteres introducidos hasta pulsar el salto de línea. Por tanto,
permite que la cadena tenga espacios en blanco*/
  printf ("La cadena es %s\n", cadena);
  printf ("El valor asignado a las variables
es:\na=%d\nb=%d\nc=%d\n", a, b, c);

  return (0);
}
```

¿Detecta algún error?
Ejecute el programa anterior asignando a cadena el valor "Hola", ¿qué sucede? Repita la ejecución asignando a cadena el valor "Buenos días", ¿qué observa?

Solución:
Al asignar a `cadena` (variable en la que se puede almacenar una cadena de como máximo 7 caracteres más el carácter nulo '/0') el valor "Buenos días" (se necesitaría una cadena de longitud 12), se sobreescriben algunas variables y se puede generar una excepción.