

PRÁCTICAS PARA LA ASIGNATURA

FUNDAMENTOS DE INFORMÁTICA

PRÁCTICA 3:

Tipos de datos y expresiones en C.

INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Pedro María Alcover Garau
Dolores Cano Gil
Carlos Fernández Andrés
Pedro Javier Navarro Lorente



PRÁCTICA TERCERA

1. El siguiente programa calcula y visualiza el Área de un triángulo. Para ello solicita por teclado la base y la altura. Trabaja con variables de distinto tipo.

```
#include <stdio.h>

void main (void)
{
    int iAltura,iBase,iArea;
    float fArear,fAreai;

    printf("\n introducir la base de un triangulo: ");
    scanf("%d", &iBase);
    printf("\n introducir la altura de un triangulo: ");
    scanf("%d", &iAltura);

    iArea=iBase*iAltura/2;          /* resultado entero*/
    fArear=(float)iBase*iAltura/2; /* resultado real*/
    fAreai=iBase*iAltura/2;

    printf("\n el área del triángulo es %d", iArea);
    printf("\n el área del triangulo es %f", fAreai);
    printf("\n el área del triangulo es %f", fArear);
}
```

2. Realizar un programa en lenguaje C para Convertir grados Fahrenheit a Celsius: $C=(5/9)*(F-32)$.

Advertencia: recuerde que la operación entre dos enteros da un resultado que es otro entero del mismo tipo.

3. El siguiente programa trabaja con la situación de overflow en variables de tipo *char*. Justifica el resultado que visualiza por pantalla.

```
#include<stdio.h>
void main(void)
{
    char a = 127;
    a++;
    printf("%hd", a);
}
```



Práctica nº 3: Tipos de datos y expresiones en C. **3 de 5**

4. Realiza otros programas como el anterior, modificando los tipos de datos (*long int*, *short int*, *float*, *double*).

Recuerde que el rango del tipo de dato **int** quedará determinado por el compilador. Según en cuál se trabaje, una variable **int** se comportará como una variable **long**, y según dónde, como una variable **short**.

5. El siguiente programa determina los tamaños en bytes de los diferentes tipos de datos en C.

```
#include <stdio.h>
void main (void)
{
    printf("\nTamaño de un entero en este compilador: ");
    printf("%ld bytes.", sizeof(int));
    printf(" y en bits = %ld", sizeof(int) * 8);

    printf("\nTamaño de un entero corto: ");
    printf("%ld bytes.", sizeof(short int));

    printf("\nTamaño de un entero largo: ");
    printf("%ld bytes.", sizeof(long int));

    printf("\nTamaño de un float: ");
    printf("%ld bytes.", sizeof(float));

    printf("\nTamaño de un double: ");
    printf("%ld bytes.", sizeof(double));

    printf("\nTamaño de un long double: ");
    printf("%ld bytes.", sizeof(long double));
}
```

6. Ejercicio de expresiones con variables tipo char.

```
#include <iostream.h>

void main (void)
{
    char a='X',b='Y';

    printf("\n valor caracter de a = %c", a);
    printf(" y en decimal = %d", (int)a);

    printf("\n valor caracter de b = %c", b);
    printf(" y en decimal = %d", (int)b);
}
```



Práctica nº 3: Tipos de datos y expresiones en C. **4 de 5**

```
printf("\n valor caracter a + b = %c", (char) (a + b));  
printf(" y en decimal = %d", (int) (a + b));  
  
printf("\n valor caracter a+b+5 = %c", (char) (a+b+5));  
printf(" y en decimal = %d", (int) (a + b + 5));  
  
printf("\nvalor caracter a+b+'5'=%c", (char) (a+b+'5'));  
printf(" y en decimal = %d", (int) (a + b + '5'));  
}
```

7. Evaluar las expresiones del siguiente programa, indicando cual es la respuesta en cada caso.

```
#include <stdio.h>  
  
void main(void)  
{  
    int x,y,z,m,n;  
    float r;  
  
    x=5;  
    y=2;  
    z=4;  
    r=0.5;  
    m=1;  
    n=2;  
  
    printf("\n expresion 1 es = %hd", (x-2>z&&!r==m/n));  
    printf("\n expresion 2 es = %hd", (r*x>y ||y*r*x<=z));  
}
```

8. Realizar la traza del siguiente programa indicando cuales serian los resultados visualizados por pantalla:

```
#include <stdio.h>  
  
void main(void)  
{  
    const float pi=3.141592;  
    int x,y,z;  
    int expr;  
    float e;  
  
    e=2.7182;  
    x=1;  
    y=4;  
    z=10;
```



Práctica nº 3: Tipos de datos y expresiones en C. **5 de 5**

```
expr = (x > 3 && (y == 4 || x + y <= z))
printf("\nexpr 1 = %d", expr);
expr = (!(2 * x == y / 2 || (pi - e) * z > y));
expr = expr != y / 2 == 2 * x && y < (pi - e) * z;
printf("\nexpr 2 = %d", expr);
}
```

Puede ver más ejercicios planteados y resueltos en el Manual "Fundamentos de Informática. Programación en Lenguaje C."