Tipos de datos

Tamaño y signo

Tipos de datos (de mayor a menor)

Tipo	printf	scanf
long double	`%Lf`	`%Lf`
double	`%f`	`%lf`
float	`%f`	`%f`
unsigned long int	`%lu`	`%lu`
long int	`%ld`	`%ld`
unsigned int	`%u`	`%u`
int	`%d`	`%d`
unsigned short int	`%hu`	`%hu`
short int	`%hd`	`%hd`
unsigned char	`%u`	`%u`
short	`%hd`	`%hd`
char	`%c`	`%c`

Función sizeof

EjSizeOf.c

- **Sintaxis**

  sizeof(nombre de la variable)

- **Ejemplo**

```
Un caracter ocupa 1 byte
Un entero   ocupa 4 bytes
Un flotante ocupa 4 bytes
Un doble    ocupa 8 bytes
```

```
char car;
int entero;
float flotante;
double doble;

printf("Un caracter ocupa %d byte\n", sizeof(car));
printf("Un entero   ocupa %d bytes\n", sizeof(entero));
printf("Un flotante ocupa %d bytes\n", sizeof(flotante));
printf("Un doble    ocupa %d bytes\n", sizeof(doble));
```

Conversion_int_char.c

```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("\n  ucarac    = %u    %c\n", ucarac, ucarac);
  printf("  caracter = %d    %c\n", caracter, caracter);
  printf("  uentero  = %u\n", uentero);
  printf("  entero   = %d\n", entero);

  entero = -2147483649;
  printf("\n  entero   = %d\n", entero);

  return 0;
}
```

Que imprime?

Conversion_int_char.c

```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("ucarac   = %u     %c\n", ucarac, ucarac);
  printf("caracter = %d     %c\n", caracter, caracter);
  printf("uentero  = %u\n", uentero, uentero);
  printf("entero   = %d\n", entero, entero);

  entero = -2147483649;
  printf("entero   = %d\n", entero, entero);

  return 0;
}
```

Que imprime?

Conversion_int_char.c

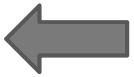
```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("ucarac   = %u    %c\n", ucarac, ucarac);
  printf("caracter = %d    %c\n", caracter, caracter);
  printf("uentero  = %u\n", uentero, uentero);
  printf("entero   = %d\n", entero, entero);

  entero = -2147483649;
  printf("entero   = %d\n", entero, entero);

  return 0;
}
```



-128 + (250-127)-1= **-6**

Conversion_int_char.c

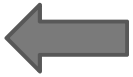
```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("ucarac   = %u     %c\n", ucarac, ucarac);
  printf("caracter = %d     %c\n", caracter, caracter);
  printf("uentero  = %u\n", uentero, uentero);
  printf("entero   = %d\n", entero, entero);

  entero = -2147483649;
  printf("entero   = %d\n", entero, entero);

  return 0;
}
```



4294967295 - 6 + 1 =
4294967290

Conversion_int_char.c

```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("ucarac   = %u    %c\n", ucarac, ucarac);
  printf("caracter = %d    %c\n", caracter, caracter);
  printf("uentero  = %u\n", uentero, uentero);
  printf("entero   = %d\n", entero, entero);

  entero = -2147483649;
  printf("entero   = %d\n", entero, entero);

  return 0;
}
```

```
ucarac   = 250      ·
caracter = -6       ·
uentero  = 4294967290
entero   = -6
```

Conversion_int_char.c

```c
#include <stdio.h>
int main()
{ unsigned int uentero; /* [0, 4294967295] */
  int entero;           /* [-2147483648, 2147483647] */
  char caracter;        /* [-128, 127] */
  unsigned char ucarac; /* [0, 255] */

  ucarac = 250;
  caracter = ucarac;
  uentero = caracter;
  entero = caracter;

  printf("ucarac   = %u     %c\n", ucarac, ucarac);
  printf("caracter = %d     %c\n", caracter, caracter);
  printf("uentero  = %u\n", uentero, uentero);
  printf("entero   = %d\n", entero, entero);

  entero = -2147483649;
  printf("entero   = %d\n", entero, entero);

  return 0;
}
```

Es el valor mínimo -1

```
entero    = 2147483647
```

Conversión entre tipos de datos

```c
/* Tipos de datos nuevos */
int main()
{
    int a, b, c; /* Entero, de -32768 a 32767 sin punto decimal */
    char x, y, z; /* De -128 a 127 sin punto decimal */
    float numero, gato, casa; /* De 3.4E-38 a 3.4E+38 con punto decimal

    a = b = c = -27;
    x = y = z = 'A';
    numero = gato = casa = 3.6792;
    a = y; /* a es ahora 65 (caracter A) */
    x = b; /* x es ahora -27 */
    numero = b; /* num será -27.00 */
    a = gato; /* a tomará el valor de 3 */

    return 0;
}
/* Este programa no muestra nada, :-) */
```

- El tipo de dato **char** es casi igual al entero excepto que solo se le pueden asignar valores entre -128 y 127 (estos valores dependen del tamaño en bytes).

- Al asignar un **float** a un **int** el valor se trunca al entero menor.

Ejemplo11.c

Conversion_int_char2.c

```c
#include <stdio.h>
int main()
{ unsigned int uentero;
  char caracter;
  unsigned char ucarac;

  uentero = -190 ;
  caracter = uentero;
  ucarac = uentero;

  printf("entero    = %u  %d\n", uentero, uentero);
  printf("caracter = %d  %c\n", caracter, caracter);
  printf("ucarac    = %u  %c\n", ucarac, ucarac);
  return 0;
}
```

Que imprime?

```
entero    = 4294967106  -190
caracter = 66  B
ucarac    = 66  B
```

Ejercicio

- Escriba un programa que lea una secuencia de letras terminada en '.' y la imprima en mayúsculas.

Ejercicio

- Escriba un programa que lea una secuencia de letras terminada en '.' y la imprima en mayúsculas.

EjercTiposdeDatos.c

```
#include <stdio.h>
int main()
{
    char car;
    scanf("%c",&car);
    while (car != '.'){
        printf("%c",((car>='a')&(car<='z'))? car-='a'-'A':car);
        scanf("%c",&car);
    }
    return 0;
}
```

Si cambiamos **char** por **int** funciona?

– o + ?