

Entrada y Salida en C++

Fundamentos de Programación

Curso: 1º



Introducción



- Flujo de entrada: serie de entradas que alimentan un ordenador para que el programa las utilice.
- Flujo de salida: serie de salidas que el programa genera.
- Suponemos que:
 - Entrada estándar: teclado.
 - Salida estándar: pantalla



Salidas con cout

- Se puede enviar a la pantalla cualquier combinación de variables y cadenas:

```
cout << num_dulces << "dulces\n";
```

```
cout << num_dulces;  
cout << "dulces" << endl;
```

- Se pueden incluir expresiones aritméticas:

```
cout << "El precio total es:" << (precio1 + precio2);
```

Operador de inserción

FP

3



Salidas con cout

- Secuencias de caracteres especiales: todas ellas comienza con el carácter '\
- Algunas secuencias de caracteres:

<code>\n</code>	Nueva línea
<code>\t</code>	Tabulación horizontal
<code>\\</code>	Diagonal invertida
<code>\"</code>	Comillas dobles

- Ejemplo:

```
//enviar línea en blanco (pantalla  
cout << "\n";
```

4



Formateo de salidas con funciones de flujo

- Formato: organización de la salida de un programa en C++ es su formato.
- Métodos asociados a los flujos de salida:

- **precision** : Fija el número de cifras después de la coma.

```
cout.precision(2);
```

- **width**: indica el número de caracteres que debe usar al enviar un elemento a la salida.

- **fill**: indica el carácter que completa los espacios en blanco cuando se manda un elemento a la salida.

```
cout.fill('a');
```

Solo afecta
Siguiete salida
Por pantalla

FP

5



Formateo de salidas con funciones de flujo

- **setf**: (set flags) establece banderas. El efecto sobre el flujo depende de la bandera.

<code>ios::fixed</code>	Formato de punto fijo
<code>ios::showpoint</code>	Incluya un punto decimal en números en coma flotante.
<code>ios::showpos</code>	Signo + para números positivos
<code>ios::left</code>	Siguiente número en el extremo izquierdo.
<code>ios::right</code>	Siguiente número en el extremo derecho.

FP

6



Formateo de salidas con funciones de flujo

- **unsetf**: cualquier bandera establecida se puede desactivar con esta función miembro.
- Ejemplos:

La bandera queda fija todo el programa

```
cout.setf(ios::fixed);  
cout.setf(ios::showpos);  
cout.setf(ios::showpoint);  
cout.setf(ios::left);  
cout.setf(ios::right);
```

Activa

```
cout.unsetf(ios::showpos);
```

Desactiva

Ejemplo con función precision



```
#include <iostream.h>  
#include <stdlib.h>  
  
int main()  
{  
    float minumero;  
  
    minumero = 3.7345;  
  
    cout.precision(3);  
    cout << "El numero ( si se toman 3 digitos) es: "  
        << minumero << endl;  
  
    cout.setf(ios::fixed);  
    cout.setf(ios::showpoint);  
    cout.precision(3);  
    cout << "El numero ( si se toman 3 cifras después del punto) es: "  
        << minumero << endl;  
  
    system("PAUSE");  
    return 0;  
}
```

Resultado de la ejecución



```
E:\clases\bdi\Curso02-03\laboratorio\practical\nada.exe
El numero < si se toman 3 digitos> es: 3.73
El numero < si se toman 3 cifras despúes del punto> es: 3.734
Presione una tecla para continuar . . . _
```

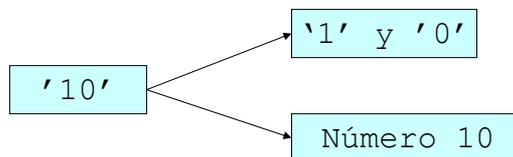
Entrada con cin



- `cin` es el flujo de entrada estándar (supondremos teclado).
Ejemplo:

```
cin >> num_pasteles;
```

- Todos los datos se envían como caracteres. La interpretación depende del programa:



Entrada y salida de caracteres



- **Funciones miembro o métodos:**

- **get:** permite leer un carácter de entrada y guardarlo en una variable tipo carácter (char).

```
char siguiente_simbolo;  
cin.get(siguiente_simbolo);
```

- **put:** es análoga a la función miembro get sólo que se emplea para salida. Se envía a la salida un carácter.

```
char siguiente_simbolo='a';  
cout.put(siguiente_simbolo);
```

- **ignore:** ignora num caracteres mientras no se encuentre el carácter delim

```
cin.ignore(int num, int delim);  
cin.ignore(); //ignora un caracter
```

FP

11

Entrada y salida de caracteres



- **Comparación entre cin >> y cin.get:**

cin.get

```
char c1,c2,c3,c4;  
  
//Introduce 4 caracteres separados por espacios  
cin.get(c1);  
cin.get(c2);  
cin.get(c3);  
cin.get(c4);  
  
//Imprimo los caracteres leidos  
cout << "Los cuatro caracteres leidos son:"<< endl;  
cout.put(c1);  
cout.put(c2);  
cout.put(c3);  
cout.put(c4);
```

12

Entrada y salida de caracteres



- Comparación entre `cin >>` y `cin.get`:

`cin >>`

```
char c1,c2,c3,c4;
//Prueba con cin
//Introduce 4 caracteres separados por espacios
cin >> c1;
cin >> c2;
cin >> c1;
cin >> c2;

//Imprimo los caracteres leidos
cout << "Los cuatro caracteres leidos son:<< endl;
cout << c1 << c2 << c3 << c4;
```

FP

13

Comparación



```
CA E:\clases\FundaProg\curso02-03\ejemplos\entrada.exe
Introduce cuatro caracteres separados por espacios
1 2 3 4
Los cuatro caracteres leidos son:
1 2 Presione una tecla para continuar . . . _
```

```
CA E:\clases\FundaProg\curso02-03\ejemplos\entrada2.exe
Introduce 4 caracteres separados por espacios
1 2 3 4
Los cuatro caracteres leidos son:
1234
Presione una tecla para continuar . . . _
```

4

Ejercicio final:



- Escribir un programa en C++ que pida dos fechas y calcule la diferencia en segundos entre ellas.

```
#include <iostream.h>
#include <stdio.h>

int main()
{
    int dd_i, mm_i, aa_i;
    int dd_f, mm_f, aa_f;
    int dd_dif, mm_dif, aa_dif;
    int num_segundos;
    char caracter;

    cout <<
        "Introducir fecha inicial(dd mm aa): " ;
    cin >> dd_i >> mm_i >> aa_i;
    cout << endl <<
        "Introducir fecha final (dd mm aa): " ;
    cin >> dd_f >> mm_f >> aa_f;

    //Calculo del total en segundos
    num_segundos = ((aa_f - aa_i)*365*24*60*60) +
        ((mm_f - mm_i)*31*24*60*60) +
        ((dd_f - dd_i)*24*60*60);

    //Calculo la diferencia en dias/meses/anyos

    aa_dif= num_segundos / (365*24*60*60);

    num_segundos = num_segundos % (365*24*60*60);

    mm_dif = num_segundos / (30*24*60*60);

    num_segundos = num_segundos % (30*24*60*60);
    dd_dif = num_segundos / (24 * 60 * 60);
```

Año = 365 días

Mes = 30 días

Día = 24 horas

```
// Salida ...
cout << endl;
cout << "\tDia\t\tMes\t\tAnyo" << endl;
cout << "\t---\t\t---\t\t---" << endl;

cout << "\t" << dd_i << "\t\t" << mm_i <<
    "\t\t" << aa_i << endl;
cout << "\t" << dd_f << "\t\t" << mm_f <<
    "\t\t" << aa_f << endl;
cout << "\t---\t\t---\t\t---" << endl;

cout << "\t" ;
cout.width(2); cout.fill('0');
cout << dd_dif << " dias" ;
cout << "\t\t" ;

cout.width(2);
cout.fill('0');
cout << mm_dif << " meses" ;

cout << "\t" ;
cout.width(2); cout.fill('0');
cout << aa_dif << " anyos" << endl;

cout << endl << "\t Transcurridos " <<
num_segundos << " segundos" << endl;
cout << endl << endl << " \t\t... pulsa
para salir" ;

return 0;
}
```

C:\Documents and Settings\Administrator\Desktop\fechas.exe

Introducir fecha inicial <dd mm aa>: 1 1 01

Introducir fecha final <dd mm aa>: 16 10 02

Dia	Mes	Año
1	1	1
16	10	2
15 días	09 meses	01 años

Transcurridos 56937600 segundos

-- pulsa para salir

Ejercicio: justificar las columnas de números a la derecha

FP

17

```
#include <iostream.h>
#include <stdlib.h>
//Escribir programa que lea tres números reales y los sume y muestro el resultado
con tres decimales.
int main()
{
    float a, b, c, res;

    cout << "Introduce 3 números reales para sumar:\n ";
    cin >> a >> b >> c;
    res = a + b + c;

    //Formateo la salida con 2 decimales
    cout.precision(3);
    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.setf(ios::right);
    cout.width(8); cout.fill(' '); cout << a << endl;
    cout.width(8); cout.fill(' '); cout << b << endl;
    cout.width(8); cout.fill(' '); cout << c << endl;
    cout<<"_____"<<endl;
    cout.width(8); cout.fill(' '); cout << res << endl;

    return 0;
}
```