

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <iostream></code>	<code>cin</code>	Flujo de entrada estándar (teclado). Es un objeto de la clase « <code>istream</code> ».
	<code>>></code>	Operador de extracción de datos con formato de un flujo de entrada.
	<code>cout</code>	Flujo de salida estándar (pantalla). Es un objeto de la clase « <code>ostream</code> ».
	<code><<</code>	Operador de inserción con formato de datos en un flujo de salida
	<code>endl</code>	Inserta en el flujo el contenido del búfer temporal, lo vacía e inserta en el flujo un fin de línea.
	<code>flush</code>	Inserta en el flujo el contenido del búfer temporal y lo vacía.
	<code>left</code>	Los datos a insertar con formato en un flujo de salida son alineados a izquierda (inserta los caracteres de relleno a la derecha del dato).
	<code>right</code>	Los datos a insertar con formato en un flujo de salida son alineados a derecha (inserta los caracteres de relleno a la izquierda del dato).
	<code>boolalpha</code>	Los datos de tipo <code>bool</code> con formato son escritos mediante secuencias de caracteres: <code>true/false</code> .
	<code>noboolalpha</code>	Los datos de tipo <code>bool</code> con formato son escritos como datos enteros: 0 o 1.
	<code>fixed</code>	Establece la notación fija como notación por defecto para escribir números reales con formato.
	<code>scientific</code>	Establece la notación científica como notación por defecto para escribir números reales con formato.
<code>#include <iomanip></code>	<code>setw(int c)</code>	Fija el número mínimo de caracteres al escribir el siguiente dato, completando, en caso necesario con caracteres de relleno.
	<code>setfill(char c)</code>	Define « <code>c</code> » como carácter de relleno.
	<code>setprecision(int n)</code>	En notación fija, determina el número de decimales, « <code>n</code> » al escribir números reales. En notación científica determina el número de cifras significativas a escribir.

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <cmath></code>	<code>double sin(double a)</code>	Devuelve $\sin a$ [el ángulo «a» en radianes]
	<code>double cos(double a)</code>	Devuelve $\cos a$ [el ángulo «a» en radianes]
	<code>double tan(double a)</code>	Devuelve $\tan a$ [el ángulo «a» en radianes]
	<code>double exp(double x)</code>	Devuelve e^x
	<code>double log(double x)</code>	Devuelve $\log_n x$
	<code>double log10(double x)</code>	Devuelve $\log_{10} x$
	<code>double log2(double x)</code>	Devuelve $\log_2 x$
	<code>double sqrt(double x)</code>	Devuelve la raíz cuadrada de «x»
	<code>double pow(double x, double y)</code>	Devuelve x^y

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <cctype></code>	<code>char toupper(char c)</code>	Si «c» es una letra minúscula del alfabeto inglés, devuelve la mayúscula correspondiente; si no lo es, devuelve el propio «c».
	<code>char tolower(char c)</code>	Si «c» es una letra mayúscula del alfabeto inglés, devuelve la minúscula correspondiente; si no lo es, devuelve el propio «c».
	<code>bool isupper(char c)</code>	Devuelve true si el valor del argumento «c» es una letra mayúscula del alfabeto inglés. En caso contrario, devuelve false .
	<code>bool islower(char c)</code>	Devuelve true si el valor del argumento «c» es una letra minúscula del alfabeto inglés. En caso contrario, devuelve false .
	<code>bool isalpha(char c)</code>	Devuelve true si el valor del argumento «c» es una letra del alfabeto inglés (mayúscula o minúscula). En caso contrario, devuelve false .
	<code>bool isdigit(char c)</code>	Devuelve true si el valor del argumento «c» es un dígito decimal ('0', '1', '2', '3', '4', '5', '6', '7', '8' o '9'). En caso contrario, devuelve false .
	<code>bool isalnum(char c)</code>	Devuelve true si el valor del argumento «c» es una letra del alfabeto inglés (mayúscula o minúscula) o un dígito decimal. En caso contrario, devuelve false .
	<code>bool isblank(char c)</code>	Devuelve true si el valor del argumento «c» es un espacio en blanco (' ') o un carácter de tabulación (representado en C++ como '\t'). En caso contrario, devuelve false .
	<code>bool isspace(char c)</code>	Devuelve true si el valor del argumento «c» es un espacio en blanco (' '), un tabulación horizontal ('\t'), un carácter de nueva línea ('\n'), un tabulador vertical ('\v'), un carácter de avance de página ('\f') o un carácter de retorno de carro ('\r'). En caso contrario, devuelve false .
<code>bool ispunct(char c)</code>	Devuelve true si el valor del argumento «c» es un signo de puntuación ('!', '"', '#', '\$', '%', '&', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\\', ']', '^', '_', '`', '{', ' ', '}' o '~'). En caso contrario, devuelve false .	

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula include	Recursos predefinidos visibles	Observaciones
#include <cstring>	void strcpy(char copia[], char original[])	Asigna a «copia» la cadena de caracteres que almacena «original», incluido el carácter nulo (en el que se detiene).
	void strcat(char cad[], char cad2[])	Asigna a «cad» la cadena de caracteres resultante de concatenar a la cadena almacenada inicialmente en «cad» la cadena «cad2». El carácter nulo con el que finaliza «cad» es sobrescrito por el primer carácter de «cad2» e incluye un carácter nulo al final de la cadena formada por la concatenación de «cad» con «cad2».
	void strcmp(char cad1[], char cad2[])	Compara carácter a carácter las cadenas «cad1» y «cad2» comenzando desde el índice 0; devuelve 0 si el contenido de ambas es idéntico, devuelve un valor positivo si en la primera discrepancia entre ambas el carácter de «cad1» tiene un código ASCII mayor que el de «cad2» y devuelve un valor negativo si en la primera discrepancia entre ambas el carácter de «cad1» tiene un código ASCII menor que el de «cad2».
	int strlen (char cad[])	Devuelve el número de caracteres de la cadena «cad» (sin contar el carácter nulo final).

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <fstream></code>	<code>istream</code>	Clase cuyos objetos gestionan un flujo de entrada.
	<code>ifstream</code>	Clase cuyos objetos gestionan un flujo de entrada asociado a un fichero. Los objetos de la clase « <code>ifstream</code> » lo son también de la clase « <code>istream</code> ».
	<code>ostream</code>	Clase cuyos objetos gestionan un flujo de salida.
	<code>ofstream</code>	Clase cuyos objetos gestionan un flujo de salida asociado a un fichero. Los objetos de la clase « <code>ofstream</code> » lo son también de la clase « <code>ostream</code> ».
	<code>fstream</code>	Clase cuyos objetos gestionan un flujo de entrada y salida asociado a un fichero.
	<code>f.open(char nombreFichero[])</code>	Asocia el fichero de nombre « <code>nombreFichero</code> » al flujo « <code>f</code> », objeto de tipo « <code>ifstream</code> », « <code>ofstream</code> » o « <code>fstream</code> ».
	<code>f.is_open()</code>	Devuelve <code>true</code> si y solo si el flujo « <code>f</code> » tiene asociado un fichero. « <code>f</code> » es un objeto de la clase « <code>ifstream</code> », « <code>ofstream</code> » o « <code>fstream</code> ».
	<code>f.close()</code>	Disocia del flujo « <code>f</code> » el fichero que estaba asociado a él y lo libera. « <code>f</code> » es un objeto de la clase « <code>ifstream</code> », « <code>ofstream</code> » o « <code>fstream</code> ».
	<code>f << expresión</code>	Operador de inserción (escritura) con formato del valor de <i>expresión</i> en el flujo « <code>f</code> », que es de la clase « <code>ostream</code> ».
	<code>f >> v</code>	Operador de extracción (lectura) con formato de un dato del flujo « <code>f</code> » para asignarlo a la variable « <code>v</code> ». « <code>f</code> » es un objeto de la clase « <code>istream</code> ».
<code>int f.get()</code>	Extrae un carácter del flujo « <code>f</code> » y devuelve el valor del carácter extraído. « <code>f</code> » es un objeto de la clase « <code>istream</code> ».	

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <fstream></code>	<code>f.get(char& c)</code>	Extrae un carácter del flujo «f» y lo asigna a «c». «f» es un objeto de la clase «istream».
	<code>f.getline(char sec[], streamsize n)</code>	Extrae una secuencia de caracteres del flujo «f» hasta finalizar la línea y los asigna a «sec». Extrae el final de la línea de «f» y añade un carácter nulo al final de «sec». Como máximo, extrae un total de «n» caracteres, contando el delimitador. streamsize es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «istream».
	<code>f.getline(char sec[], streamsize n, char delimitador)</code>	Extrae una secuencia de caracteres del flujo «f» hasta que encuentra la primera aparición del carácter «delimitador», que también es extraído de «f», asigna a «sec» los caracteres extraídos, excepto el «delimitador» y añade un carácter nulo al final de «sec». Como máximo, extrae un total de «n» caracteres, contando el delimitador. streamsize es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «istream».
	<code>f.get(char sec[], streamsize n)</code>	Extrae una secuencia de caracteres del flujo «f» hasta finalizar la línea y los asigna a «sec». No extrae el final de la línea de «f», pero sí que añade un carácter nulo al final de «sec». Como máximo, extrae un total de «n» – 1 caracteres. streamsize es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «istream».

Recursos predefinidos en C++ que son utilizados en la asignatura

Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <fstream></code>	<pre>f.get(char sec[], streamsize n, char delimitador)</pre>	<p>Extrae una secuencia de caracteres del flujo «f» hasta que encuentra la primera aparición del carácter «delimitador», que no es extraído. Asigna a «sec» los caracteres extraídos y añade un carácter nulo al final de «sec». Como máximo, extrae un total de «n» – 1 caracteres. streamsize es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «istream».</p>
	<pre>f.put(char c)</pre>	<p>Inserta el carácter «c» en el flujo «f», que es un objeto de la clase «ostream».</p>
	<pre>f.write(char sec[], streamsize n)</pre>	<p>Inserta la secuencia de caracteres almacenada en «sec», hasta un máximo de «n» caracteres, en el flujo «f». streamsize es un tipo entero, dependiente del compilador. «f» es un objeto de la clase «ostream».</p>
	<pre>f.eof()</pre>	<p>Devuelve true si y solo si la última operación de extracción de información no pudo completarse por no haber datos extraíbles del fichero asociado al flujo «f», que es un objeto de la clase «istream».</p>
	<pre>f.read(reinterpret_cast<char*>(&v), sizeof(T))</pre>	<p>Extrae del flujo «f» (objeto de la clase «ifstream» asociado a un fichero binario) un dato de tipo «T» y lo asigna a la variable «v».</p>
	<pre>f.write(reinterpret_cast<const char*>(&v), sizeof(T))</pre>	<p>Inserta en el flujo «f» (objeto de la clase «ofstream» asociado a un fichero binario) el valor de la variable «v» de tipo «T».</p>

Recursos predefinidos en C++ que son utilizados en la asignatura		
Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
	<code>f.seekg(streampos pos)</code>	Indica al flujo «f» que realice la siguiente operación de lectura a partir del <i>byte</i> indicado por la posición «pos». streampos es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «istream».
	<code>f.seekp(streampos pos)</code>	Indica al flujo «f» que realice la siguiente operación de escritura a partir del <i>byte</i> indicado por la posición «pos». streampos es un tipo entero sin signo, dependiente del compilador. «f» es un objeto de la clase «ostream».

Recursos predefinidos en C++ que son utilizados en la asignatura		
Cláusula <code>include</code>	Recursos predefinidos visibles	Observaciones
<code>#include <cstdlib></code>	<code>int atoi(char sec[])</code>	Analiza la secuencia de caracteres «sec» (ej.: <code>sec = "-127"</code>), como la descripción textual de un número entero y devuelve el valor entero correspondiente (ej: <code>-127</code>)
	<code>double atof(char sec[])</code>	Analiza la secuencia de caracteres «sec» (ej.: <code>sec = "-1.475"</code>), como la descripción textual de un número real y devuelve el valor real correspondiente (ej: <code>-1,475</code>).