Ejercicio 1:

Crea un programa que calcule la resta entre 2016 y 1974 y la multiplicación de 78 y -19. No utilices variables. Se tendrá en cuenta la presentación de los datos por pantalla. Guardalo como constantes.cpp

Ejercicio 2:

Crea un programa que sume dos variables las cuales las hemos inicializado previamente a los valores 10 y 8 respectivamente. Guardalo como suma1.cpp

Ejercicio 3:

Modifica el programa anterior para que pida dos números al usuario y nos de como resultado la suma. Guardalo como suma2.cpp

Ejercicio 4:

En fichero ***cuadradator.cpp*** hemos escrito algunas líneas y hemos comentado otras en las cuales te indican que es lo que tiene que hacer el programa o la fórmula que ha de hacer el algoritmo. Abre el programa ***cuadradator.cpp*** quita los comentarios que se te indican y pon la instruciones correspondientes. Guardalo como cuadrado.cpp

Ejercicio 5:

Escribe un programa donde el usuario nos introduce un número de segundos y nos los transforma en h:m:s. Guardalo como hora.cpp
La salida ha de ser:

```
Por favor, escriba el número de segundos: 8600

8600 segundos son 2h 23m 20s
```

Ejercicio 6:

Crea un programa que pida al usuario una cantidad en km y la transforme a milla terrestres (1 mill = 1,609 km). Usa variables del tipo float o double. Al principio del programa haz un comentario explicando que hace el programa. ¿Qué diferencia encuentras si utilizas variables float o double? (escríbelo como comentario dentro del fichero .cpp). Guarda como millas.cpp

Ejercicio 7:

Escribe un programa que pida al usuario las notas de tres exámenes (estas notas podrán llevar decimales) y nos calcule la media. Guarda como media_notas.cpp

Ejercicio 8:

Crea un programa que pida un numero entero al usuario e indique se es par o no. Para ello, deberá comprobar si el resto que obtiene al dividir dicho número entre dos es cero: ***if(numero%2==0).*** Guardalo como es_par.cpp

Ejercicio 9:

Realiza un programa que pida al usuario dos números enteros y diga cuál es mayor. Guardalo como es_mayor.cpp.

Ejercicio 10:

Realiza un programa que pida la edad al usuario y que si esa edad está entre 0 y 12 años diga que ***"eres un niñ@"***, si esta entre 13 y 18 diga que **"eres un adolescente"**, si esta entre 19 y 70 ***"eres un adulto"***, si es mayor de 71 que ponga que ***"es un ancian@"*** y en otro caso que ***"esa edad no es posible"***. Guardalo con el nombre edad.cpp.
Utiliza la estructura:

```
If (condición)
{
        ...
}
else if (condición)
{
        ...
}
(tantos else if como necesite)
else
{
        ...
}
```

Ejercicio 11:

Utilizando la instrucción ***switch*** escribe un programa que pida al usuario el número de mes estamos y nos escriba por pantalla el mes que corresponda a ese número. Guardalo con el nombre meses.cpp.

Ejercicio 12:

La instrucción ***switch*** se utiliza mucho para elaborar un menú de opciones, donde el usuario debe introducir el número de la opción deseada. Realiza un programa con el siguiente menú.

1. Leer el horóscopo estándar.
2. Calcula la media de dos números.
3. Salir

Introduce una opción:

Si introduces la opción 1, mostrarás:
HOROSCOPO: Tal y como se prevee, tendrás un finde movidito. OJO: Si te pasas bebiendo puedes amanecer con alguien que de miedo...(y no soy yo).
Si introduces la opción 2 deberá hacer:
solicita dos números y te saca la media
Si introduces la opción 3, te mostrará:
Ha seleccionado salir jueves, viernes y sábado... naaaaaaaaa.
Para cualquier otra opción:
Opción incorrecta. María invita en la cantina

Guardalo con el nombre menu.cpp

Ejercicio 13

Utilizando ***while***, haz un programa que pida al usuario al usuario la contraseña tantas veces como sea necesario hasta que introduzca el número 7879. Guardalo con el nombre contraseña.cpp

Ejercicio 14

Haza un programa que pida al usuarios números enteros y los vaya sumando mientras y cuando introduzca un número negativo pare y nos de el resultado de la suma de todos ellos. Resuelve el programa utilizando un ***do... while(condición).*** Explica entre comentarios que has hecho para que no te sume el número negativo. Guardalo con el nombre do_while_suma.cpp.
El programa deberá se:

Salida programa

```
-------Si desea salir escriba un número negativo-------
Escribe un número: 10
-------Si desea salir escriba un número negativo-------
Escribe un número: 5
-------Si desea salir escriba un número negativo-------
Escribe un número: 10
-------Si desea salir escriba un número negativo-------
Escribe un número: -1
La suma ha sido: 25
```

Ejercicio 15

Escribe un programa que escriba todos lo números pares desde cero (incluido) hasta el cien. Utiliza un contador. ***for(inicialización;condición;incremento).*** Guardalo con el nombre pares.cpp

Salida programa

```
Estos son los números pares desde 0 a 100
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66
68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
```

Ejercicio 16

Haz lo mismo que el programa anterior pero ahora con los números impares desde cero (sin incluirlo) hasta el 100. Guardalo con el nombre impares.cpp

Salida programa

```
Estos son los números impares desde 0 a 100
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67
69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

Ejercicio 17

Haz un programa que pida al usuario un número cual quiera y el progrma nos muestres los números pares desde 0 hasta ese número. Guardalo con el nombre pares2.cpp

Salida programa

```
Introduzca un número positivo: 10
Estos son los números pares desde 0 a 10
0 2 4 6 8 10
```

Rúbricas

A continuación se indican las rúbricas que se siguen para corregir esta práctica.

	Muy bien (10)	Bien (7)	Regular (4)	Mal (0)
Entrega	Ha entregado en tiempo y por la vía demandada la práctica.			No la entregado en tiempo o por otra vía que no es la demandada.
Compila	Compila el programa sin ningún error.			Da algún error de compilación
Ejecuta	Se ejecuta el programa y la salida por pantalla es como se pide o en su defecto es clara y ordenada.	Se ejecuta el programa aunque la salida no exactamente igual a la que se pide o no es clara y ordenada.	Se ejecuta el programa aunque de una manera caótica sin que sea nada clara la secuencia de ejecución.	No se ejecuta el programa correctamente.
Código en general	El código es muy claro y fácil de leer; utilizando espacios, tabulaciones, la llaves se visualizan muy bien, etc	El código es muy claro y fácil de leer aunque tiene algún fallo de tabulación o llaves, etc	El código no es muy claro.	El código no es claro.
Instrucciones a aplicar para resolver el ejercicio	Ha utilizado la/las instrucciones correctamente como se esperaba para la resolución del ejercicio.	Ha utilizado casi todas las instrucciones correctamente.	Ha utilizado algunas instrucciones.	No ha utilizado ninguna instrucción correctamente para la resolución del ejercicio.
Algoritmo	Resuelve el problema eficazmente	Resuelve el problema pero es mejorable.	Resuelve el problema pero es poco eficaz.	No resuelve el problema
Comentarios	Utiliza comentarios para definir el programa y para explicar la instrucciones más importantes del algoritmo.	Utiliza comentarios para definir el programa.	Utiliza algunos comentarios.	No utiliza comentarios.
Nombre de los ficheros	El nombre es igual que el que se solicita, mismo caracteres, mayúsculas y minúsculas	El nombre es igual que el que se solicita aunque no tiene la mismas mayúsculas y minúsculas	El nombre no es el mismo pero sí es representativo para el contenido del mismo.	Otro caso.