

## Práctica 3: Estructuras de control condicionales (2 sesiones)

### Programación 1. Grado en Ingeniería Multimedia

26 de septiembre de 2016

#### Objetivos:

- Revisar el concepto de algoritmo y entender la necesidad del diseño de algoritmos en el estudio y resolución de programas.
- Comprender y diferenciar perfectamente los distintos tipos de instrucciones utilizadas en el diseño de algoritmos.
- Conocer y manejar con habilidad las sentencias de control condicionales.

1.

Diseña un programa que pida un número por teclado y nos diga si es par o impar, es decir, si es divisible por dos o no.

2.

Modifica el programa 6 de la práctica 2 para que además de calcular el IMC dado el peso y la altura informe al paciente de su estado de salud de acuerdo a la siguiente tabla.

IMC	Salud
Menor que 16	Falto de Peso
De 16 a 25	Normal
Mayor que 25	Sobrepeso

3.

Realiza un programa en el que a partir de 3 números enteros solicitados por teclado nos diga cual es el mayor, cual es el menor y cual es el de en medio.

4.

Implementa un algoritmo que lea un carácter e indique si es o no una letra. Además, en caso de ser letra, debe indicarse si es vocal o consonante y si está en minúsculas o mayúsculas. Ten en cuenta que en lenguaje C la comparación de caracteres utiliza el código ASCII asociado a cada carácter. Sugerencia: buscar en Internet o en un libro “tabla códigos ASCII”.

5.

Implementa un programa que “juegue” a una versión muy simplificada del juego de cartas de las siete y media. En este juego el objetivo es que las cartas sumen el valor más cercano posible a 7.5 sin pasarse. El programa pedirá por teclado los valores de dos cartas, entre el 1 y el 12. Cada carta cuenta como su valor excepto las figuras (10, 11 y 12) que cuentan como medio punto. El programa debería imprimir al final si se podría pedir otra carta (el valor de las dos es menor de 7.5), si deberíamos plantarnos (7.5) o si ya nos hemos pasado (>7.5). Para simplificar no tengas en cuenta que el usuario podría introducir números no válidos para las cartas.

6.

Tres empresas de acceso a Internet han lanzado diferentes ofertas de tarifa plana parcial mensual para esta temporada de verano. Dichas empresas proponen una tarifa plana de lunes a viernes entre las 15:00 y las 8:00, que también se aplica para el fin de semana. Para el resto, es decir, de lunes a viernes del 8:00 a 15:00, ofrecen una tarificación adicional por minuto. Por tanto, el coste total mensual será la suma de la tarifa plana más los minutos utilizados con el horario adicional. Los importes de las diferentes compañías son los siguientes:

Empresa/Tarifa	Tarifa plana	Tarifa minuto
PapaFon / C1	16 euros	0.03 euros
Movestor / C2	15 euros	0.06 euros
Tevés / C3	11 euros	0.05 euros

Escribe un programa que permita determinar qué oferta de esas tres nos resultaría más económica en función de los minutos de conexión del usuario fuera del horario de la tarifa plana. Para ello, el programa deberá solicitar el número de minutos que el usuario accede al servicio de lunes a viernes entre las 8:00 y las 15:00 durante un mes. Como salida el programa debe proporcionar el coste

mensual de cada compañía, el coste mensual más económico y el nombre de la compañía que lo ofrece.

7.

Se pretende gestionar el cobro por fracciones horarias en un parking. Se cobra por fracciones de 30 minutos, donde cada una de ellas cuesta 0.35 euros.

Por ejemplo, para un tiempo de permanencia en el parking de 17 minutos, la cantidad a abonar sería de 0.35 euros. Observa que si se ha estado menos de 30 minutos, se abona la fracción completa. De modo similar para un tiempo de 91 minutos habría que abonar el importe correspondiente a 4 fracciones, es decir, 1.40 euros.

Realiza dos versiones del programa:

- A partir de la fecha, hora y minuto de entrada y la fecha, hora y minuto de salida indique la cantidad a pagar. (Se debe comprobar que la hora de salida es superior a la hora de entrada. Puede permanecer en el parking más de un día)
- A partir de la fecha, hora y minuto de entrada indique la cantidad a pagar. (La hora y fecha de salida se debe extraer del sistema)

#### ANEXO (extracción fecha y hora del sistema)

```
#include<iostream>
#include<ctime>
using namespace std;

int main() {
    time_t fecha_sistema, hora_local;
    fecha_sistema = time(NULL);
    hora_local = time (NULL);

    struct tm *tiempo = localtime(&fecha_sistema);
    struct tm *horario = localtime(&hora_local);

    int dia, mes, anio;
    int hora,min;

    hora = horario->tm_hour;
    min = horario->tm_min;

    anio=tiempo->tm_year + 1900;
    mes=tiempo->tm_mon + 1;
    dia=tiempo->tm_mday;

    cout<< dia << "/" << mes << "/" << anio <<endl;
    cout << hora << ":" << min << endl;

    return 0;
}
```