

1. (2 puntos) Escribe en lenguaje C un programa para visualizar en pantalla el triángulo de Floyd de límite num. Para ello se debe imprimir de forma secuencial los números naturales: 1, 2, 3, ..., num en líneas sucesivas, colocando **un** número en la **primera** línea, **dos** en la **segunda**, **tres** en la **tercera**, y así sucesivamente. Por ejemplo, para num =12 obtendremos el siguiente triángulo:

1			
2	3		
4	5	6	
7	8	9	10
11	12		

En la codificación suponer que num se introduce de manera correcta. Para un alineado correcto (hasta 9999) cada número ocupará cuatro espacios, tal y como se ilustra en el ejemplo anterior.

```
#include <stdio.h>

int main (void)
{
    int n; /* Límite */
    int i; /* Contador de números */
    int l; /* Contador de línea */
    int j; /* Contador de número en esta línea */

    printf ("Introduce el límite: ");
    scanf ("%d", &n);
    i = 1;
    j = 1;
    l = 1;
    while (i <= n) {
        printf ("%4d", i);
        i++;
        j++;
        if (j > l) {
            printf ("\n");
            l++;
            j = 1;
        }
    }
    return 0;
}
```

2. (4 puntos) Escribir un **programa C** que pida una frase y la convierta a lenguaje de mensajes SMS, escribiéndola a continuación en pantalla.

Las normas para convertir la frase a SMS son las siguientes:

- Cambiar la subcadena "qu" por "k".
- Sustituir las subcadenas "por" o "par" por "x".
- Quitar todas las vocales de la frase menos la 'a'. Para ello escribir y utilizar una **función** que dado un carácter nos diga si es una vocal o no.

Se supone que la frase inicial está en minúsculas y que puede contener un máximo de 255 caracteres.

**Ejemplo:** Si la frase inicial leída es `cad1`:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...
q	u	e	d	a	m	o	s		y		p	a	s	o		p	o	r		t	u	\0	...

La cadena con el resultado del mensaje SMS `cad2` será:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...
k	d	a	m	s		y		p	a	s		x		t	\0	...	...	...	...	...	...	...	...

**Ejemplo de ejecución:**

Dime una frase: quedamos a las 3 y paso por tu casa para ver la peli

Mensaje SMS: *kdams a las 3 y pas x t casa xa vr la pl*

**Nota:** el texto subrayado del ejemplo es el introducido por teclado. Se ha resaltado en cursiva el contenido del mensaje SMS de la cadena resultante `cad2`.

```
#include <stdio.h>

int es_vocal (int car);

int main (void)
{
    char fra[256];
    char sms[256];
    int i, j;

    printf ("Introduce la frase: ");
    gets (fra);

    for (i = j = 0; fra [i] != '\0'; i++) {
        if (fra[i] == 'q' && fra[i+1] == 'u') {
            sms[j++] = 'k';
            i++; /* Adelantamos la 'u' ya leída */
        }
        else if (fra[i] == 'p' &&
            (fra[i+1] == 'o' || fra[i+1] == 'a') &&
            fra[i+2] == 'r') {
            sms[j++] = 'x';
            i += 2; /* Adelantamos "or" o "ar" ya leídos */
        }
        else if (!es_vocal (fra[i]) || fra[i] == 'a')
            sms[j++] = fra[i];
        /* Si no ignoramos el carácter, es decir, no lo compiamos */
    }
    sms[j] = '\0';
    printf ("Frase: [%s]\nSms: [%s]\n", fra, sms);

    return 0;
}

int es_vocal (int car)
{
    return car == 'a' || car == 'e' || car == 'i' || car == 'o' || car == 'u';
}
```

3. (1 punto) Completa los huecos de la siguiente tabla con las equivalencias en las respectivas bases, utilizando los dígitos necesarios en cada caso. Describe escuetamente el método utilizado.

Base 2	Base 4	Base 7	Base 10	Base 16
11001	121	34	25	19
10011000	2120	305	152	98
101100	230	62	44	2c

4. (3 puntos) Disponemos de las funciones:

diaSistema	Devuelve día, mes y año del reloj del sistema (de hoy)
diaJul	Devuelve una fecha en formato numérico (juliano)
diaGrg	Convierte de formato numérico a formato día-mes-año
diaSem	Devuelve el día de la semana (0 - 6) de una fecha juliana

Sus prototipos, especificados en el fichero "**fechas.h**" son los siguientes:

```
void diaSistema (int *dd, int *mm, int *aa);
long diaJul (int dd, int mm, int aa);
void diaGrg (long jul, int *dd, int *mm, int *aa);
int diaSem (long jul);
```

a) Escribe una **función** que escriba la fecha de mañana (según la fecha del sistema) de la siguiente manera:

Mañana es martes, 19 de Septiembre de 2006

b) Escribe un **programa** que llame a la función del apartado anterior para mostrar la fecha de mañana.

```
#include <stdio.h>
#include "fechas.h"

void manana (void);
char *txtDiaSem (int jul);
char *txtMes (int mes);

int main (void)
{
    manana ();
    return 0;
}

void manana (void)
{
    int dia, mes, ano; /* fecha en formato Gregoriano (día/mes/año) */
    long fl;           /* fecha en formato Juliano (numérico) */

    diaSistema (&dia, &mes, &ano);
    fl = diaJul (dia, mes, ano);
    fl++;
    diaGrg (fl, &dia, &mes, &ano);

    printf ("Mañana es %s, %d de %s de %d\n",
            txtDiaSem(diaSem(fl)), dia, txtMes(mes), ano);
}
```

```
char *txtDiaSem (int jul)
{
    switch (diaSem (jul)) {
        case 0: return "lunes";
        case 1: return "martes";
        case 2: return "miercoles";
        case 3: return "jueves";
        case 4: return "viernes";
        case 5: return "sabado";
        case 6: return "domingo";
        default: return "error";
    }
}

char *txtMes (int mes)
{
    switch (mes) {
        case 1: return "Enero";
        case 2: return "Febrero";
        case 3: return "Marzo";
        case 4: return "Abril";
        case 5: return "Mayo";
        case 6: return "Junio";
        case 7: return "Julio";
        case 8: return "Agosto";
        case 9: return "Septiembre";
        case 10: return "Octubre";
        case 11: return "Noviembre";
        case 12: return "Diciembre";
        default: return "error";
    }
}
```