Programación III
Mobile Numeric Keypad Problem
(Memoization/Tabulation)

1. Nombre y apellido de los miembros del grupo.

   - Victoria María Montelongo Martín
   - Ricardo Arbelo Guillén

2. Dirección de correo de los miembros del grupo.

   - victoria.montelongo101@alu.ulpgc.es
   - ricardo.arbelo101@alu.ulpgc.es

3. Enunciado del problema.

   Este problema busca conocer las posibles combinaciones de N números que se puedan hacer con el teclado numérico de un teléfono móvil convencional. Como condiciones, sólo se puede pulsar los botones que están arriba, izquierda, derecha o abajo del botón actual. En adición, no se permite pulsar los botones de las esquinas de la fila inferior (*"*"* y *"#"*).

4. Referencias a páginas web.

   - Página con una lista de problemas de diversos niveles de dificultad:
     https://www.geeksforgeeks.org/dynamic-programming/

   - Página de dónde sacamos la información para entender el problema escogido:
     https://www.geeksforgeeks.org/mobile-numeric-keypad-problem/

   - Páginas web con código que nos ayudaron a llegar a la solución presentada:
     https://www.geeksforgeeks.org/mobile-numeric-keypad-problem/
     https://www.tutorialspoint.com/Mobile-Numeric-Keypad-Problem
     https://www.geeksforgeeks.org/mobile-numeric-keypad-problem-set-2/

5. Recurrencia.

Con respecto al <u>contador</u> de combinaciones:

t(i, j, length)
- 0 : **length == 0 || Se presione la tecla "*" || Se presione la tecla "#" || Se salga por fuera del teclado**
- 1 : **length == 1**
- t(i, j, length-1) + t(i-1, j, length-1) + t(i+1, j, length-1) + t(i, j-1, length-1) + t(i, j+1, length-1)

Con respecto a las <u>combinaciones</u>:

t(i, j, length)
- [""] : **length == 0 || Se presione la tecla "*" || Se presione la tecla "#" || Se salga por fuera del teclado**
- [keyPad[i][j]] : **length == 1**
- list = [ ]
  for combination in t(i, j, length-1): list.append(keyPad[i][j] + combination)
  for combination in t(i-1, j, length-1): list.append(keyPad[i-1][j] + combination) **(Sólo si las combinaciones de t(i-1, j, length-1)) son válidas)**
  for combination in t(i+1, j, length-1): list.append(keyPad[i+1][j] + combination) **(Sólo si las combinaciones de t(i+1, j, length-1)) son <u>válidas</u>)**
  for combination in t(i, j-1, length-1): list.append(keyPad[i][j-1] + combination) **(Sólo si las combinaciones de t(i, j-1, length-1)) son <u>válidas</u>)**
  for combination in t(i, j+1, length-1): list.append(keyPad[i][j+1] + combination) **(Sólo si las combinaciones de t(i, j+1, length-1)) son <u>válidas</u>)**

6. Explicación de la recurrencia.

Con respecto a la recurrencia del contador de combinaciones, cada vez que se pulse una tecla válida, dentro de cada combinación, se suma un uno al contador.

Con respecto a la recurrencia de las combinaciones, para todas las combinaciones válidas resultantes de presionar el mismo botón, el de arriba, el de abajo, el de la izquierda y el de la derecha, se le añade el valor de la tecla presionada (keyPad[i][j]) al principio de todas éstas.

7. Análisis asintótico.

A continuación, realizaremos en análisis asintótico de nuestro código en Python, tanto de la solución utilizando *memoization*, como de la *tabulation*.

<u>Memoization</u>

Primero, analizaremos el press_key, ya que el orden de una llamada a un subprograma es igual al orden del subprograma llamado. Al ser un algoritmo recursivo para sacar la recurrencia debemos estudiar la parte iterativa y la recursiva y al ser todo operaciones elementales, la parte iterativa tiene complejidad 1 mientras que la parte recursiva reduce en cada llamada el rango en uno. Por tanto, la recurrencia obtenida es:

$T(n) = 5T_{n-1} + 1$

con condición inicial:

$T(1) = 1$

La solución obtenida Wolfram Alpha:

Quitando las constantes, nos quedaría un análisis de orden O(5^n).

Como en el memoization tenemos dos bucles con O(1) porque no dependen de n y una llamada recursiva, el análisis del memoization daría como resultado O(5^n).

<u>Tabulation</u>

Para desarrollar el análisis asintótico identificamos la operación crítica que se repetirá más veces en el algoritmo, que en este caso es el *if* de la línea 103, que se encuentra dentro de tres *for.* Todas las operaciones dentro de este triple bucle son de orden 1, incluida la operación crítica, por otro lado, tenemos el bucle:

```
for k in range(2, length+1):
    for i in range(1, 5):
        for j in range(1, 4):
```

Podemos observar que el segundo y tercer *for* son de Orden 1, pero el primero depende de "*length*", por tanto, decimos que es n, y como resultado el orden sería:

O(1*1*n) = O(n)

8. Copia de pantalla que muestre el uso del programa Python desde consola activando combinaciones de todos los switches obligatorios.

   - py MobileNumericKeypadProblem.py

```
C:\Users\Ricardo>cd C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python

C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python>py MobileNumericKeypadProblem.py

You must specify an argument. If help is needed, type -h or --help when executing.
```

- py MobileNumericKeypadProblem.py -h

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python>py MobileNumericKeypadProblem.py -h

Optional arguments:
-------------------------------------------------------------------------------------------
Short argument          Long argument             Explanation
-------------------------------------------------------------------------------------------
-h                      --help                    Shows this message and exit.

-d [DIRECTORY]          --directory [DIRECTORY]   directory (proccess many files).
-f [FILE]               --file [FILE]             file (proccess a single file).

-sm                     --memoization             Solve it with Memoization.
-st                     --tabulation              Solve it with Tabulation.
-check                                            Check whether Memoization and Tabulation got the same results.

-t                      --time                    Display time.
-nd                     --numberDigits            Display the number of digits (length).
-nc                     --numberCombinations      Display the number of possible combinations.
-dc                     --displayCombinations     Display the possible combinations.
```

- py MobileNumericKeypadProblem.py -f data/data_0.txt -sm -t -nd -nc -dc

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python>py MobileNumericKeypadProblem.py -f data/data_0.txt -sm -t -nd -nc -dc
The elapsed time is:                        0.0 seconds
The number of digits per combination is:    0
The total amount of possible combinations is:  0
The possible combinations are:
There are no keys to be pressed (Count: 0)
```

- py MobileNumericKeypadProblem.py -f data/data_1.txt -st -t -nd -nc -dc

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python>py MobileNumericKeypadProblem.py -f data/data_1.txt -st -t -nd -nc -dc
The elapsed time is:                        0.0 seconds
The number of digits per combination is:    1
The total amount of possible combinations is:  10
The possible combinations are:
If we start with 0, valid numbers will be: 0 (Count: 1).
If we start with 1, valid numbers will be: 1 (Count: 1).
If we start with 2, valid numbers will be: 2 (Count: 1).
If we start with 3, valid numbers will be: 3 (Count: 1).
If we start with 4, valid numbers will be: 4 (Count: 1).
If we start with 5, valid numbers will be: 5 (Count: 1).
If we start with 6, valid numbers will be: 6 (Count: 1).
If we start with 7, valid numbers will be: 7 (Count: 1).
If we start with 8, valid numbers will be: 8 (Count: 1).
If we start with 9, valid numbers will be: 9 (Count: 1).
```

- py MobileNumericKeypadProblem.py -d data -check

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Python>py MobileNumericKeypadProblem.py -d data -check

Analizando el fichero data\data_0.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_1.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_2.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_3.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_4.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_5.txt
Tabulation and Memoization both obtain the same results!

Analizando el fichero data\data_6.txt
Tabulation and Memoization both obtain the same results!
```

9. Copia de pantalla que muestre el uso del programa Java desde consola activando combinaciones de todos los switches obligatorios.

- java MobileKeypad

```
C:\Users\Ricardo>cd C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java

C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>javac MobileKeypad.java
Note: MobileKeypad.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad

You must specify an argument. If help is needed, type -h or --help when executing.
```

- java MobileKeypad --help

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad --help

Optional arguments:
-------------------------------------------------------------------------------------------
Short argument          Long argument              Explanation
-------------------------------------------------------------------------------------------
-h                      --help                     Shows this message and exit.

-d [DIRECTORY]          --directory [DIRECTORY]    directory (proccess many files).
-f [FILE]               --file [FILE]              file (proccess a single file).

-sm                     --memoization              Solve it with Memoization.
-st                     --tabulation               Solve it with Tabulation.
-check                                             Solve it with Tabulation.

-t                      --time                     Display time.
-nd                      --numberDigits            Display the number of digits (length).
-nc                      --numberCombinations      Display the number of possible combinations.
-dc                     --displayCompinations      Display the possible combinatons.
```

- java MobileKeypad -f data/data_2.txt --memoization -t -nd -nc -dc

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad -f data/data_2.txt --memoization -t -nd -nc -dc
The elapsed time is:                    0.0 seconds.
The number of digits per combination is:    2
Te total amount of possible combinations is: 36
The possible combinations are:
If we start with 0, valid number will be: 00 08 (Count: 2)
If we start with 1, valid number will be: 11 14 12 (Count: 3)
If we start with 2, valid number will be: 22 25 21 23 (Count: 4)
If we start with 3, valid number will be: 33 36 32 (Count: 3)
If we start with 4, valid number will be: 44 41 47 45 (Count: 4)
If we start with 5, valid number will be: 55 52 58 54 56 (Count: 5)
If we start with 6, valid number will be: 66 63 69 65 (Count: 4)
If we start with 7, valid number will be: 77 74 78 (Count: 3)
If we start with 8, valid number will be: 88 85 80 87 89 (Count: 5)
If we start with 9, valid number will be: 99 96 98 (Count: 3)
```

- java MobileKeypad -f data/data_3.txt --tabulation -t -nd -nc -dc

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad -f data/data_3.txt --tabulation -t -nd -nc -dc
The elapsed time is:                    0.001 seconds.
The number of digits per combination is:    3
Te total amount of possible combinations is: 138
The possible combinations are:
If we start with 0, valid number will be: 000 008 088 085 080 087 089 (Count: 7)
If we start with 1, valid number will be: 111 114 112 144 141 147 145 122 125 121 123 (Count: 11)
If we start with 2, valid number will be: 222 225 221 223 255 252 258 254 256 211 214 212 233 236 232 (Count: 15)
If we start with 3, valid number will be: 333 336 332 366 363 369 365 322 325 321 323 (Count: 11)
If we start with 4, valid number will be: 444 441 447 445 411 414 412 477 474 478 455 452 458 454 456 (Count: 15)
If we start with 5, valid number will be: 555 552 558 554 556 522 525 521 523 588 585 580 587 589 544 541 547 545 566 563 569 565 (Count: 22)
If we start with 6, valid number will be: 666 663 669 665 633 636 632 699 696 698 655 652 658 654 656 (Count: 15)
If we start with 7, valid number will be: 777 774 778 744 741 747 745 788 785 780 787 789 (Count: 12)
If we start with 8, valid number will be: 888 885 880 887 889 855 852 858 854 856 800 808 877 874 878 899 896 898 (Count: 18)
If we start with 9, valid number will be: 999 996 998 966 963 969 965 988 985 980 987 989 (Count: 12)
```

- java MobileKeypad -f data/data_4.txt -check

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad -f data/data_4.txt -check
Tabulation and Memoization both obtain the same results!
```

- java MobileKeypad -d data -sm -t -nd -nc

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>java MobileKeypad -d data -sm -t -nd -nc

Analizando el fichero data_0.txt:
The elapsed time is:                        0.0 seconds.
The number of digits per combination is:    0
Te total amount of possible combinations is: 0

Analizando el fichero data_1.txt:
The elapsed time is:                        0.0 seconds.
The number of digits per combination is:    1
Te total amount of possible combinations is: 10

Analizando el fichero data_2.txt:
The elapsed time is:                        0.0 seconds.
The number of digits per combination is:    2
Te total amount of possible combinations is: 36

Analizando el fichero data_3.txt:
The elapsed time is:                        0.001 seconds.
The number of digits per combination is:    3
Te total amount of possible combinations is: 138

Analizando el fichero data_4.txt:
The elapsed time is:                        0.001 seconds.
The number of digits per combination is:    4
Te total amount of possible combinations is: 532

Analizando el fichero data_5.txt:
The elapsed time is:                        0.002 seconds.
The number of digits per combination is:    5
Te total amount of possible combinations is: 2062

Analizando el fichero data_6.txt:
The elapsed time is:                        0.004 seconds.
The number of digits per combination is:    6
Te total amount of possible combinations is: 7990
```

10. Aclaraciones.

- **NOTA 1:**
  El switch que declara la solución por la cual se resolverá el problema (Tabulation o Memoization) es declarado justo después del nombre del fichero o directorio del cual se van a extraer los datos

- **NOTA 2:**
  Para chequear que las respuestas obtenidas mediante el procedimiento de Tabulation y mediante el de Memoization sean coincidentes, se declara el switch "*check"* tras la declaración del fichero o directorio con el que se va a trabajar, sin necesidad de escribir switches adicionales tras ello.

- **NOTA 3:**
  Para el correcto funcionamiento de programas en código Java, mostrados a través de la terminal de comandos, se han de realizar cierto paso previos a la compilación y ejecución de la aplicación:
  Dentro de la configuración avanzada del sistema, debemos incluir en la variable del entorno *"path"* la ruta de la carpeta *bin*, que está dentro del directorio JDK, la cual

contiene todas las librerías que nos permiten programar y ejecutar proyectos en Java.
(Apunte extra: ambos alumnos hemos realizado este paso en equipos con sistema operativo Windows)
Una vez realizado ello, se podrá compilar y ejecutar las aplicaciones de acuerdo con lo enseñado en el Punto 9.

- **NOTA 4:**
  Al compilar el programa en lenguaje Java, nos salen dos advertencias:

```
C:\Users\Ricardo\Documents\ULPGC\Curso 20-21\P3\Bloque 2\TRABAJO EVALUACIÓN\Código Java>javac MobileKeypad.java
Note: MobileKeypad.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

A pesar de que estas notas no perjudiquen el correcto funcionamiento de la aplicación, consideramos relevante el notificarlo puesto que se intentó, sin éxito alguno, corregir el código para que éstas desapareciesen.