



PROJECTO 1

EFEITOS ESPECIAIS E COORDENADAS EXCEL (Beta)

INTRODUÇÃO E OBJECTIVOS

A finalidade deste projecto passa por consolidar os conhecimentos de programação em Python adquiridos até agora, através da implementação de dois programas de natureza diferente. Um objectivo secundário deste projecto consiste em introduzir a programação orientada para sistemas recorrendo, para tal, a bibliotecas desenvolvidas por terceiros.

Na primeira parte do projecto, deve implementar um conjunto de algoritmos para produzir "efeitos especiais" em texto. Na segunda parte, pretende-se que desenvolva um utilitário para conversão de coordenadas tipo excel para coordenadas lineares de uma matriz. Alguns elementos do projecto são considerados "extra", sendo resumidos na secção **AVALIAÇÃO**. Nesta secção encontra também uma explicação sobre o que se pretende dizer com "extra" no contexto deste projecto.

PARTE I - EFEITOS ESPECIAIS

Nesta parte deve desenvolver um programa em Python para exibir o texto introduzido na linha de comandos de acordo com determinados "efeitos especiais" (ver à frente). O seu programa deve ser invocado da seguinte forma:

```
$ python3 efeitos.py palavra1 [palavra2] ... [palavraN]
```

Depois aplica cada um dos efeitos em baixo descritos ao texto que resulta da concatenação de `palavra1`, `palavra2`, etc. A leitura dos argumentos da linha de comandos deve ser feita por inspecção directa das variáveis `argc` e `argv`.

Os caracteres do texto (que resultam da concatenação de todas as palavras) devem ser exibidos individualmente. À excepção do último efeito, não pode utilizar construtores de strings, vectores, nem outros mecanismos para agrupar os caracteres.

Segue-se uma demonstração dos efeitos pretendidos. Assuma que o programa foi invocado da seguinte forma:

```
$ python3 efeitos.py jose mourinho
```

Efeito 1- Diagonal Esquerda...

J
O
S
E

M
O
U
R
I
N
H
O

Efeito 2 - Diagonal Esquerda, Palavras Invertidas...

O
H
N
I
R
U
O
M

E
S
O
J

Efeito 3 - Diagonais Cruzadas...

J J
O O
S S
E E

M M
O
U U
R R
I I
N N
H H
O O

Efeito 4 - Diagonal Direita, Palavras por Ordem Inversa...

M
O
U
R
I
N
H
O

J
O
S
E

Efeito 5 - Texto Deslizante em Ciclo ao Longo da linha (vamos assumir que o texto é ALBERTO)

A	L	B	E	R	T	O			0,5s	
	A	L	B	E	R	T	O		1,0s	
...									...	
E	R	T	O				A	L	B	4,0s
B	E	R	T	O				A	L	4,5s

Para este último efeito, assuma que cada linha tem 40 caracteres de largura e que o intervalo entre exibição de linhas (para que seja visível o efeito deslizante) é 0,5 segundos. Quando "está sob este este efeito", o

programa terminar apenas quando o utilizador o interrompe com CTRL+C.

Para "temporizar" a execução do código em Python moderno, pode utilizar a função `sleep` definida no módulo `time`. Esta função recebe um *float* com indicação do intervalo de tempo em segundos. Por exemplo, `sleep(1)` ou `sleep(1.0)` suspendem o processo durante 1s, ao passo que `sleep(1.5)` suspende durante um segundo e meio, e `sleep(0.001)` suspende durante um milissegundo. Para apagar o ecrã, sugere-se a invocação do comando `clear` em sistemas do tipo Unix (macOS, Linux, FreeBSD, etc.) ou o comando `cls` em sistemas Windows. Eis um exemplo que utiliza estes mecanismos:

```
import time
import subprocess

print("Início")
time.sleep(0.5)
subprocess.run(['clear'])      # substituir "clear" por "cls" em Windows
print("Meio segundo depois")
```

Como **extra**, o programa pode aceitar o parâmetro `-i` com o valor do intervalo de actualização em segundos. O valor por omissão deste parâmetro deve ser 1s. Eis como pode exibir ALBERTO a deslizar para a direita com um intervalo de actualização de 0.5s:

```
$ python3 efeitos.py -i 0.5 ALBERTO
```

Como segundo **extra**, pode também incorporar um intervalo nos efeitos 1 a 4. Neste caso, cada caractere deverá ser exibido após o intervalo de tempo indicado pelo parâmetro `-i` (mas não é necessário implementar a exibição em *loop*).

Exiba um separador apropriado entre todos os efeitos. Sugere-se uma pausa entre efeitos, seguida de um "apagar" de ecrã, para que seja visível o efeito em questão.

PARTE II - COORDENADAS EXCEL

Este exercício encontra-se descrito num dos laboratórios. Deve implementar tal como descrito nesse laboratório e na secção de entregas deste enunciado.

Como **extra**, investigue como produzir um executável, quer para este programa, quer para os programas descritos nas outras partes deste enunciado. O executável deverá conter tudo o que é necessário para correr o programa, incluindo o interpretador de Python. Descreva a produção do executável na secção apropriada do relatório.

[PARTE III - CIFRA DE CÉSAR]

A cifra de César é um algoritmo de encriptação que passa deslocar cada elemento de uma mensagem por um valor inteiro escolhido ao acaso. Se considerarmos uma mensagem como sendo uma sequência de bytes e se k for o valor do deslocamento, então todas as ocorrências do byte i são transformadas no byte $i + k$. O valor de k deve estar compreendido entre 0 e 255. A descodificação passa por reverter o processo.

O comando `ccypher` permite encriptar e desencriptar um ficheiro com a cifra de César. A sintaxe do comando deve ser a seguinte:

```
$ python3 ccypher.py -e [-s SHIFT] FILE
$ python3 ccypher.py -d -s SHIFT FILE
```

Com a opção `-e` o `ccypher` encripta o ficheiro dado por `FILE` utilizando para tal o deslocamento (`SHIFT`) passado para a opção `-s`. No caso de não ser indicado nenhum com a opção `-s`, o programa gera um deslocamento aleatório devendo no final exibir na saída padrão qual o valor desse deslocamento. Com a opção `-d`, o `cypher` desencripta o ficheiro `FILE`. Neste caso, o deslocamento é obrigatório. Este comando deve suportar as opções longas `--encrypt`, `--decrypt` e `--shift` como alternativa às opções curtas `-e`, `-d` e `-s`.

Para iludir possíveis intrusos, a extensão do ficheiro codificado deve ser `.pdf`. Utilize a biblioteca **docopt** para ler os argumentos da linha de comandos.

AVALIAÇÃO

No contexto deste projecto, um elemento "**extra**" é um componente ou funcionalidade cuja cotação (ver tabela em baixo) é substancialmente inferior à de outros componentes ou funcionalidades de dificuldade semelhante. É possível ter uma boa classificação neste projecto não realizando nenhum dos elementos extras.

Elemento	Cotação Máxima (0..20)
Efeito 1	2
Efeito 2	2
Efeito 3	3
Efeito 4	3
Efeito 5	3
Coordenadas excel	3

Elemento	Cotação Máxima (0..20)
Parâmetro -i	1
Intervalo efeitos 1 a 4	0.5
Produção de executável	1
Cifra de César	1.5

Como incentivo, a classificação das funcionalidades extra pode contribuir até 2 valor(es) para a nota do teste escrito e até 1 valor(es) para a nota do próximo projecto. O projecto deve ser resolvido em grupos de dois formandos. Excepcionalmente, poderá ser realizado por grupos com outras dimensões.

Deve também elaborar um relatório (ver secção **Prazo e Entregas**) que valerá no máximo 20% da cotação do projecto.

PRAZOS E ENTREGAS

O trabalho final deve ser entregue até às 23h59m do dia 28/03/2021. Um atraso de N dias na entrega levará a uma penalização dada pela fórmula $0.5 \times 2^{(N-1)}$ ($N > 0$).

Nesta data deverá entregar um **ZIP** com o seguinte:

1. Um ficheiro de código para cada parte do projecto, mais o PDF correspondente ao relatório (ver em baixo). Os ficheiros devem ser os seguintes:

Efeitos especiais: efeitos.py

Coordenadas excel: excel.py

Cifra de César: ccypher.py

Relatório: relatorio.pdf

2. Relatório em PDF, cuja estrutura e formatação deverão ser a do modelo fornecido em anexo. Em termos de formatação adapte apenas a designação da acção e o nome dos módulos. Siga as recomendações relacionadas com a elaboração de um relatório dadas pelo formador Fernando Ruela. O relatório deve incluir uma capa simples com o símbolo do IEFP, referência ao Centro de Formação de Alcântara, data, indicação do curso e da acção (eg, Técnico de Informática - Sistemas 07) e dos elementos que elaboraram o trabalho.

Em termos de conteúdo o seu relatório deve possuir as seguintes secções e anexos:

2.1.1 Introdução e Objectivos (não plagiar a deste enunciado!).

2.1.2 Desenho e Estrutura que deverá conter:

- Para cada efeito, um fluxograma com o algoritmo geral desse efeito especial. Assuma que todo o texto a exibir é uma variável de entrada do fluxograma. Aqui pode ignorar a questão da concatenação. Assuma também que esse texto é uma sequência de caracteres, tal como uma string em Python.
- Um fluxograma que descreva a conversão das colunas do formato "letras" o formato numérico (inteiro positivo começando em 0).

2.1.3 Implementação. A secção Implementação deve indicar aspectos de implementação que considere relevantes. É aqui que deve indicar a "docstring" utilizada pelo docopt (caso tenha utilizado esta biblioteca). Deve indicar para que plataforma desenvolveu e testou o seu programa, a versão do compilador e das bibliotecas utilizadas.

É também aqui que deve dar instruções sobre como "compilar" o seu programa e produzir um executável (num programa maior poderia ser pertinente criar uma secção Compilação e Instalação).

É também nesta secção que deve indicar que elementos opcionais foram implementados e como.

2.1.4 Conclusão. Além de seguir as recomendações sobre a elaboração de uma conclusão de relatório, deve também listar o que foi implementado e o que ficou por implementar.