

Este es un ejemplo de como pueden realizar sus documentos técnicos en itálica encontraran mis comentarios y los elementos que empiecen con # no los tienen que escribir, los coloque para que se puedan guiar por las secciones

DOCUMENTO TÉCNICO

Requisitos

Especificación

#RESUMEN DEL PROBLEMA (opcional)

La idea de esta sección es tener un breve resumen del problema

Encontrar la manera de encriptar un mensaje basándose en una letra de cada palabra, de tal manera que las letras formarán un mensaje con la primera letra de la primera palabra, la segunda letra de la segunda palabra y así sucesivamente. Si una palabra no tiene suficientes letras se sigue con la siguiente palabra.

#DIAGRAMA

Los diagramas se utilizaran en ejercicios que impliquen estructuras de datos

#ENTRADAS Y SALIDAS

En esta sección escribimos las condiciones de entrada y de salida del problema

Entrada:

La primera línea de entrada da el número de casos, $T(1 \leq T \leq 30)$, luego un espacio en blanco antes del primer caso. Cada caso representa un mensaje compuesto de entre 1 a 100 líneas, y cada línea es compuesta de entre 1 a 30 palabras. Cada palabra en la misma línea está separada por un espacio, cada palabra está conformada por letras del abecedario (mayúsculas y minúsculas) y tiene máximo 30 letras. Los únicos símbolos que pueden entrar son letras y espacios. Debe haber un espacio entre cada mensaje.

Salida:

Una cadena que es de la forma "Case #x" donde x es el número del caso correspondiente, y después de esto tantas líneas de texto compuestas por una palabra como líneas de palabras se introdujeron en cada caso

Diseño

Estrategia

#DESCRIPCIÓN GENERAL Y JUSTIFICACIÓN

En este espacio podrán explicar textualmente el algoritmo que desarrollaron para solucionar el problema

Por cada caso descifrar un mensaje, en el cual por cada linea de palabras, según el numero de la palabra ingresada, se toma la letra de la palabra correspondiente a ese numero, si el tamaño de la palabra es menor al numero se ignora esta palabra y se toma la siguiente, sin aumentar el numero.

La estrategia utilizada para resolver el problema es estrategia incremental, la idea es almacenar las palabras en una lista y con un temporal verificar si la palabra es tan larga como su posición, si es así añadir la letra correspondiente a la posición en la palabra, a una lista y aumentar el temporal. al final convertir la lista de letras en un Sting, juntar las letras y devolver el mensaje.

#ESTRUCTURA DE DATOS

En caso de utilizarse una estructura de datos se tiene que especificar cual y de que manera la utilizaron

#ALGORITMO

En este espacio dependiendo del algoritmo tendrán que especificar algo, en el caso de este ejemplo es una estrategia incremental por lo tanto se utilizaron invariantes y aserciones

Para el ejemplo voy a colocar el código dentro de este documento, pero cuando ustedes lo entreguen basta con el programa adjunto

En este fragmento de código se pueden ver los ciclos de las invariantes y el análisis de los algoritmos

```
from sys import stdin

def write(line):
    words = []
    index = 0
    for i in line: #Ciclo de la invariante #1
        if len(i) > index:
            words.append(i[index])
            index += 1
    words = map(str, words)
    print("".join(words))
    return index

# costo    pasos
# 1         1
# 1         1
# 1         n
# 1         n - 1
# 1         n - 1
# 1         n - 1
# 1         1
# 1         1
# 1         1
#Total O = (n), omega = 1

def main():
    rango = int(stdin.readline().strip())
    for i in range(rango): #Ciclo de la invariante #1
        print("Case #" + str(i+1) + ":")
        line = stdin.readline().strip().split(" ")
        while len(line[0]) > 0:
            write(line)
            line = stdin.readline().strip().split(" ")
    if __name__ == '__main__':
        main()

# costo    pasos
# 1         1
# 1         j
# 1         j-1
# 1         j-1
# 1         j-1
# write(n) j-1
# 1         j-1
#Total O = (jn+j) omega = 1
```

Invariante #1 (ver código)

Por cada caso se descifra un mensaje.

- Iniciación: No hay ningún caso.

- Estabilidad: Por cada numero dentro del rango establecido debe haber un caso.
- Terminación: Hubo tantos casos como el rango definido.

Invariante #2 (ver código)

Por cada palabra tan larga como index hay una letra sii $\text{index} > 0$.

- Iniciación: Hay un mensaje tan largo como palabras tan largas como el índice en la línea (0 al principio).
- Estabilidad: Por cada palabra del ciclo sii la palabra es tan larga como index se toma la letra y se añade al mensaje final.
- Terminación: No hay mas palabras de las cuales se pueda sacar un mensaje.

Casos de prueba

En esta sección deben poner distintos casos de prueba que consideren relevantes para el programa, estos casos de prueba los idean ustedes y son diferentes a los proporcionados por el problema, en el espacio de justificación va el porque consideran relevante el caso

Entrada	Justificación	Salida
	Línea vacía	
Ana	Una sola línea sencilla	a
Canas le sObran a mi compañera Teeodora ella es catadora de ratas	Dos líneas con mayúsculas de por medio	como esta
Lorem ipsum dolor sit amet consectetur adipiscing elit Sed urna lectus posuere ac congue in iaculis vel urna Suspendisse potenti Vestibulum id nulla nec magna aliquet consectetur Aliquam vel auctor	30 palabras.	Lplteseiuu

Análisis

En este espacio pueden colocar su código o simplemente especificar que se encuentra dentro de este

Temporal

Esta escrito como comentarios en el código.

Código

Al igual que el punto anterior pueden guiarse con este ejemplo

Documentación

Dentro del código.

Fuentes

Las fuentes son el código sobre el cual escribieron el documento

/arena_1_a