

Ejemplo: secuencia con errores

Considere una frase como una secuencia de caracteres válidos con MF. Las palabras y/o números presentes en la misma pueden estar separadas por espacios en blanco, salto de línea, salto de página, retorno de carro, tabuladores.

Diseñe y escriba un algoritmo que dada una frase le permita llevar a cabo las siguientes tareas:

- a) Contar la cantidad de letras minúsculas.
- b) Calcular la longitud promedio de las palabras que comienzan con minúscula.
- c) Contar la cantidad de espacios presentes.

Codifique su algoritmo en C y pruebe su programa para distintas frases.

ALGORITMO frase;

ENT: secuencia de caracteres con MF, car: carácter;

SAL: min, esp, sum, pal : entero >0; lprom : real ;

CONST: MF ' ';

Inicializar; // sum, min, esp, pal \leftarrow 0;

LEER (car);

MIENTRAS (car \neq MF)

| Saltar_y_contar_espacios

| SI (car es letra minúscula)

| | Tratamiento de palabras con minúscula

| SINO

| | Tratamiento de palabras con mayúscula

| FIN SI

| Saltar caracteres especiales excepto MF y espacio

FIN MIENTRAS

lprom \leftarrow sum/pal;

ESCRIBIR(min, lprom, esp)

PARAR.

Saltar_y_contar_espacios:

MIENTRAS (car es espacio y car \neq MF)

esp \leftarrow esp + 1

LEER(car)

FIN MIENTRAS

Saltar caracteres especiales excepto MF y espacio:

MIENTRAS (car no es carácter Y car \neq MF Y car no es espacio)

LEER (car)

FIN MIENTRAS

Tratamiento de palabras con minúscula:

pal \leftarrow pal + 1

MIENTRAS (car es letra y car \neq MF)

| SI (car es minúscula) ENTONCES

| | min \leftarrow min + 1

| | sum \leftarrow sum + 1 // contar la longitud

| | LEER(car);

FIN MIENTRAS;

Tratamiento de palabras con mayúscula:

MIENTRAS (car es letra y car \neq MF)

| SI (car es minúscula) ENTONCES

| | min \leftarrow min + 1

| LEER(car)

FIN MIENTRAS