

Función de Ackermann

La función de Ackermann es famosa por su importante papel en la teoría de computabilidad como el primer ejemplo de una función totalmente computable que no es primariamente recursiva.

Desde entonces la función se ha modificado ligeramente pero aún se usa cotidianamente. Esto es debido a su definición en terminos de una recursión extremadamente profunda; por esta razón se usa como una prueba (*benchmark*) de la capacidad de un compilador o intérprete para optimizar la recursión.

El objetivo de esta práctica es codificar la función de Ackermann expresada con dos parámetros m y n que retorna un valor entero según la siguiente definición:

$$\begin{aligned} \text{ack}(m,n) &= n+1 && \text{si } m=0 \\ \text{ack}(m,n) &= \text{ack}(m-1,1) && \text{si } m>0, n=0 \\ \text{ack}(m,n) &= \text{ack}(m-1, \text{ack}(m,n-1)) && \text{si } m,n > 0 \end{aligned}$$

m , n deben ser enteros no negativos; por esta razón podemos usar a m y/o n como parámetros que permiten terminar la ejecución del programa de prueba.

Sugerencia:

Ver función de Ackerman.

https://es.wikipedia.org/wiki/Funci%C3%B3n_de_Ackermann

Pruebas:

$\text{ack}(1, 1) = 3$

$\text{ack}(1, 4) = 6$

$\text{ack}(3, 4) = 125$

Cree el archivo `entrada.txt` como se ve en la Figura 1.

Cree el archivo `salida1.txt` como se ve en la Figura 1.

- Compile su programa:

```
$ cc -o RUN ackermann.c
```

- Ejecute su programa:

```
$ ./RUN > entrada.txt > prueba.txt
```

- Pruebe la salida de su programa:

```
$ diff salida1.txt prueba.txt
```

no debe dar ninguna salida indicando que los archivos `prueba.txt` y `salida1.txt` son idénticos.

Actividad posterior:

Modifique su programa original para crear una versión que cuente el número de veces que se invoca a la función `ack`. Esta nueva versión NO debe usar variables globales.

Para probar la nueva versión cree el archivo `salida2.txt`

Nota: formatear su archivo fuente usando el siguiente comando:

```
$ astyle -style==allman ackermann.c
```

```
Terminal - jcasteld@venus: ~/Programacion_I/L
File Edit View Terminal Tabs Help
jcasteld@venu... x jcasteld@venu... x jcasteld@venu... x
Ackermann git:Curso_1_2016 > cat entrada.txt
1 1
1 3
1 4
2 2
4 0
3 3
3 4
3 6
3 7
3 8
3 13
0 -2
Ackermann git:Curso_1_2016 > cat salida1.txt
3
5
6
7
13
61
125
509
1021
2045
65533
Ackermann git:Curso_1_2016 > cat salida2.txt
3 4
5 8
6 10
7 27
13 107
61 2432
125 10307
509 172233
1021 693964
2045 2785999
65533 2862983902
Ackermann git:Curso_1_2016 > █
```

Figura 1