**Programa 7, Método de Menores**

Para el desarrollo de este programa, se realizo exactamente el mismo proceso que en el de la burbuja para generar las 30000 cadenas aleatorias. Debido a que escribir todo eso de nuevo no es información, me iré directamente a explicar lo comprendido del método en si en su funcionamiento.

Este método establece el número mas grande o el numero mas pequeño posible dependiendo el tipo de dato que se ordenara, por ejemplo para el caso numérico el numero mas grande es 1431655766. En el ejemplo describiré nada mas lo que se necesita para hacer el ordenamiento de menor a mayor.

Suponiendo que tuviésemos los siguientes números, en su respectiva posición de 0 a 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 19167 | 15576 | 23858 | 3910 | 7636 | 10553 | 12681 | 27148 | 6496 | 22849 |

Lo que hacemos es comparar el valor que se encuentra en la localidad 0 con el numero mas grande que nosotros llamaremos menor, entonces decimos que iPadada=0

19167 es menor que 1431655766, debido a que esto es cierto sucede lo siguiente:

Menor obtiene el valor de 19167 y guardamos la localidad de este dato en ipos, por lo que ipos=0

Se vuelve a hacer el mismo procedimiento solo que aumentando la localidad de uno en uno hasta recorrerla toda. En cuanto se termine de recorrer todo el arreglo que se desea ordenar se notara que el numero menor es 6496 y su localidad en memoria es 8. Por lo que nuestras variables quedarían:

Menor=6496 iPos=8

Al terminar este proceso ya tenemos el numero mas pequeño y lo tendremos que poner en la localidad 0, que es donde comienza nuestro arreglo mediante:

iAuxiliar obtiene el valor de num[ipos]

iAuxiliar=num[8] o lo que es iAuxiliar = 6496

num[8] obtiene el valor num[ipasada]

num[8]=num[0] o lo que es igual num[8]=19167

num[ipasada] obtiene el valor de iAuxiliar

num[0]=iAuxiliar o lo que es igual num[0]=6496

Fiscamente quedaría:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 6496 | 15576 | 23858 | 3910 | 7636 | 10553 | 12681 | 27148 | 19167 | 22849 |

y Se incrementara iPasada de uno en uno, y se repetirá todo el proceso hasta que quede ordenado.. Como se puede observar el método es muy sencillo de entender.

Si se quisiera ordenar de mayor a menor se tendría que cambiar la variable menor por el numero mas pequeño que se puede representar.

Habiendo entendido todo lo anterior, solo se tiene que generar una cadena que represente el numero mas grande es decir la que repita el digito 9 dependiendo el largo de cadena o el numero mas pequeño que antecede el símbolo – y después sigue el digito 9 dependiendo el largo de la cadena.

Para hacer la comparación se utilizo la función de strcmp y para copiar las cadenas la función strcpy, que también en el programa de la burbuja fue desarrollado el uso, realmente no ha cambiado la aplicación, lo único que ha cambiado es la manera en que se ordena.

Referencias Electrónicas:

<http://msdn.microsoft.com/en-us/library/e0z9k731.aspx>

<http://msdn.microsoft.com/en-us/library/kk6xf663.aspx>

<http://www.ibm.com/developerworks/linux/library/l-u-cyr/>