Encoding Algorithms

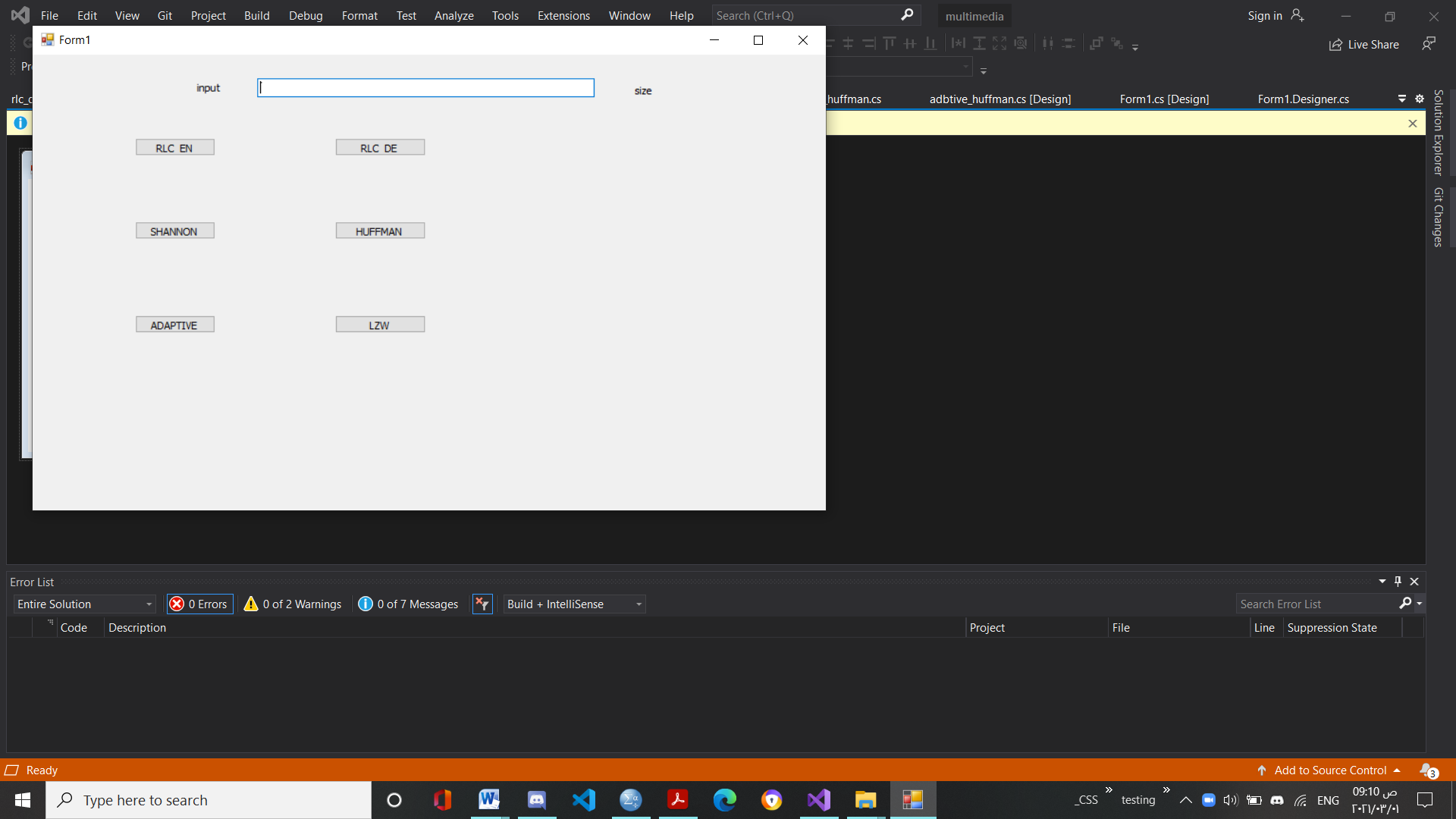
Run-Length

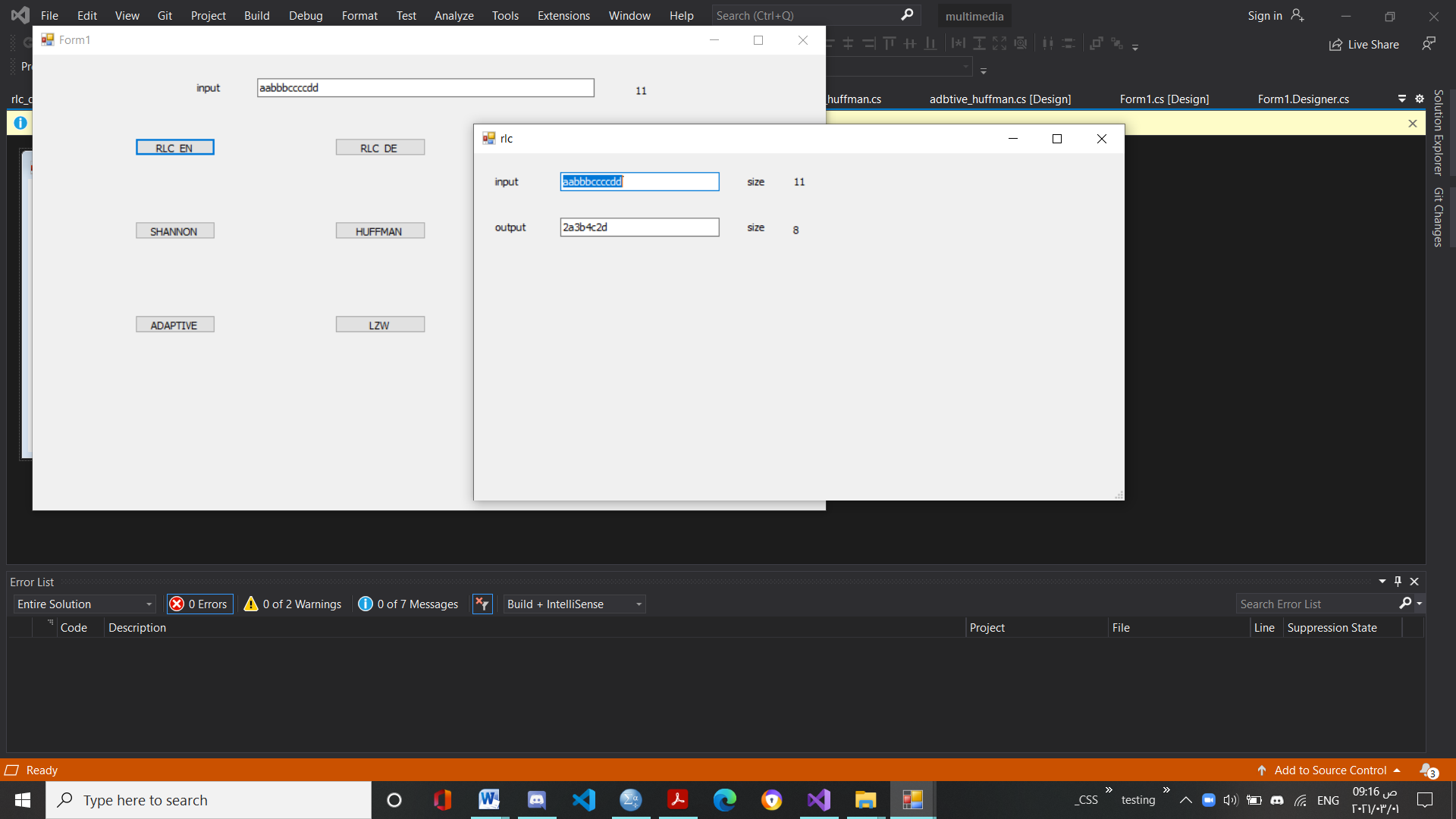
Variable Run-Length

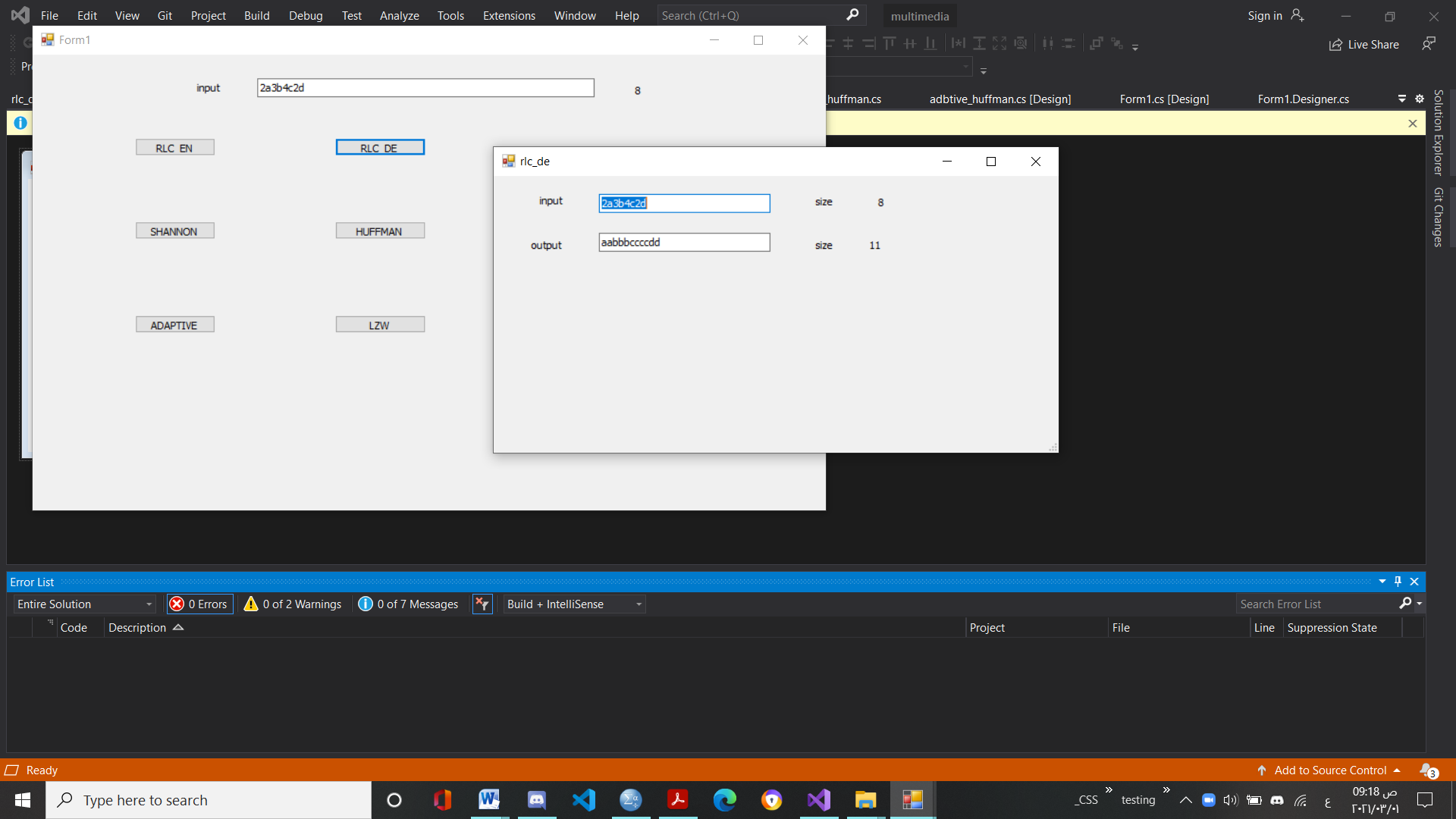
Huffman

Adaptive Huffman

LZW







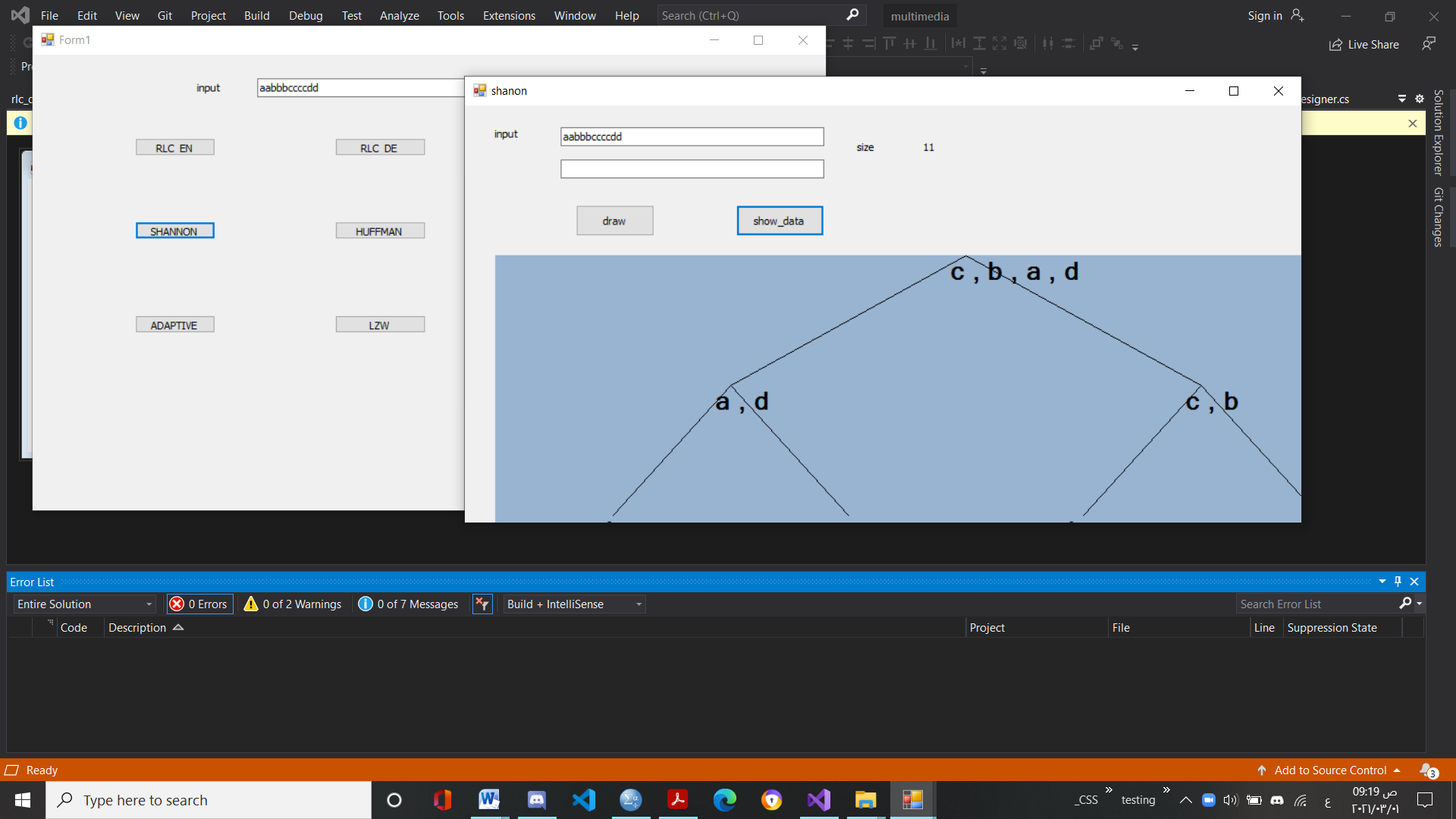
**Run-length encoding (RLE)**

This encoding method is frequently applied to graphics-type images (or pixels in a scan line) — simple compression algorithm in its own right. It is also a component used in JPEG compression pipeline. Basic RLE Approach (e.g. for images): Sequences of image elements X1, X2, . . . , Xn (row by row).

• Mapped to pairs (c1, L1),(c2, L2), . . . ,(cn, Ln), where ci represent image intensity or colour and Li the length of the i-th run of pixels.

• (Not dissimilar to zero length suppression above.)

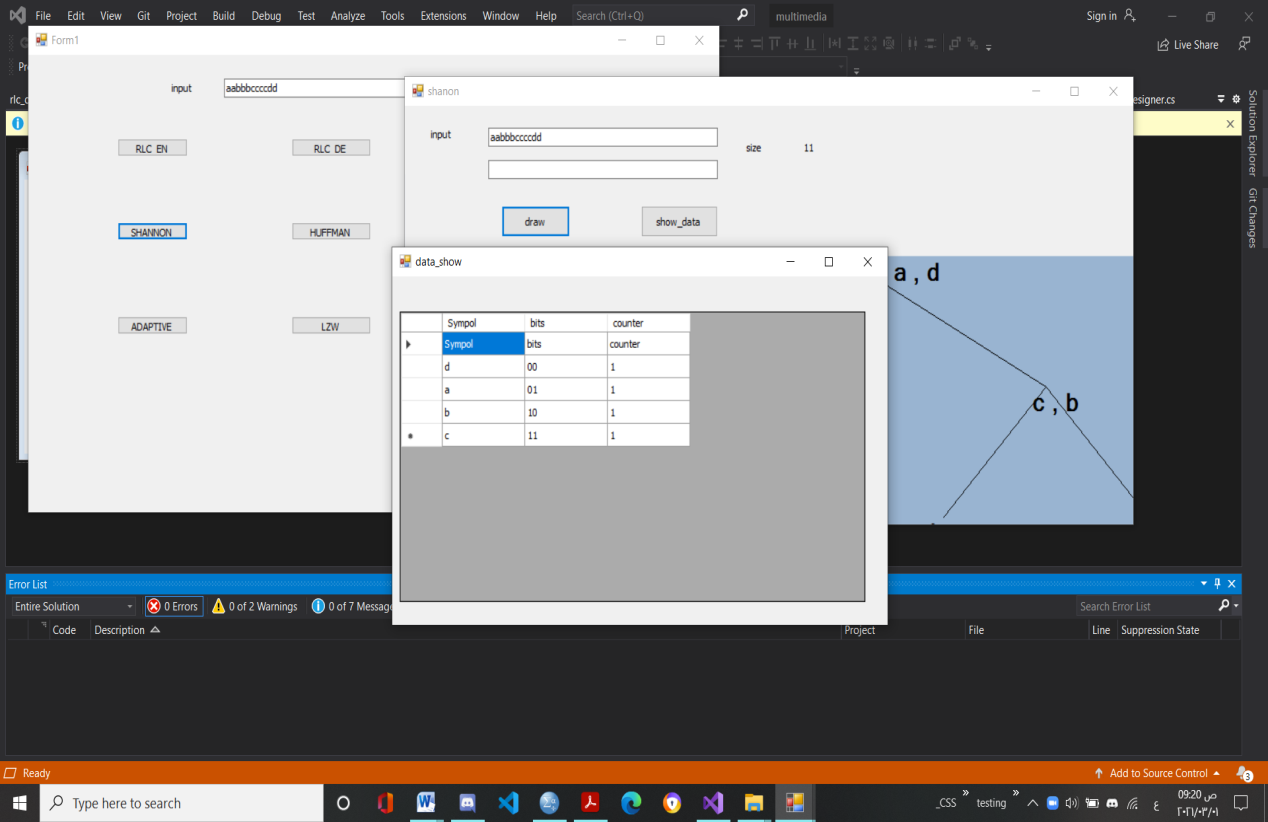
The savings are dependent on the data: In the worst case (random noise) encoding is more heavy than original .



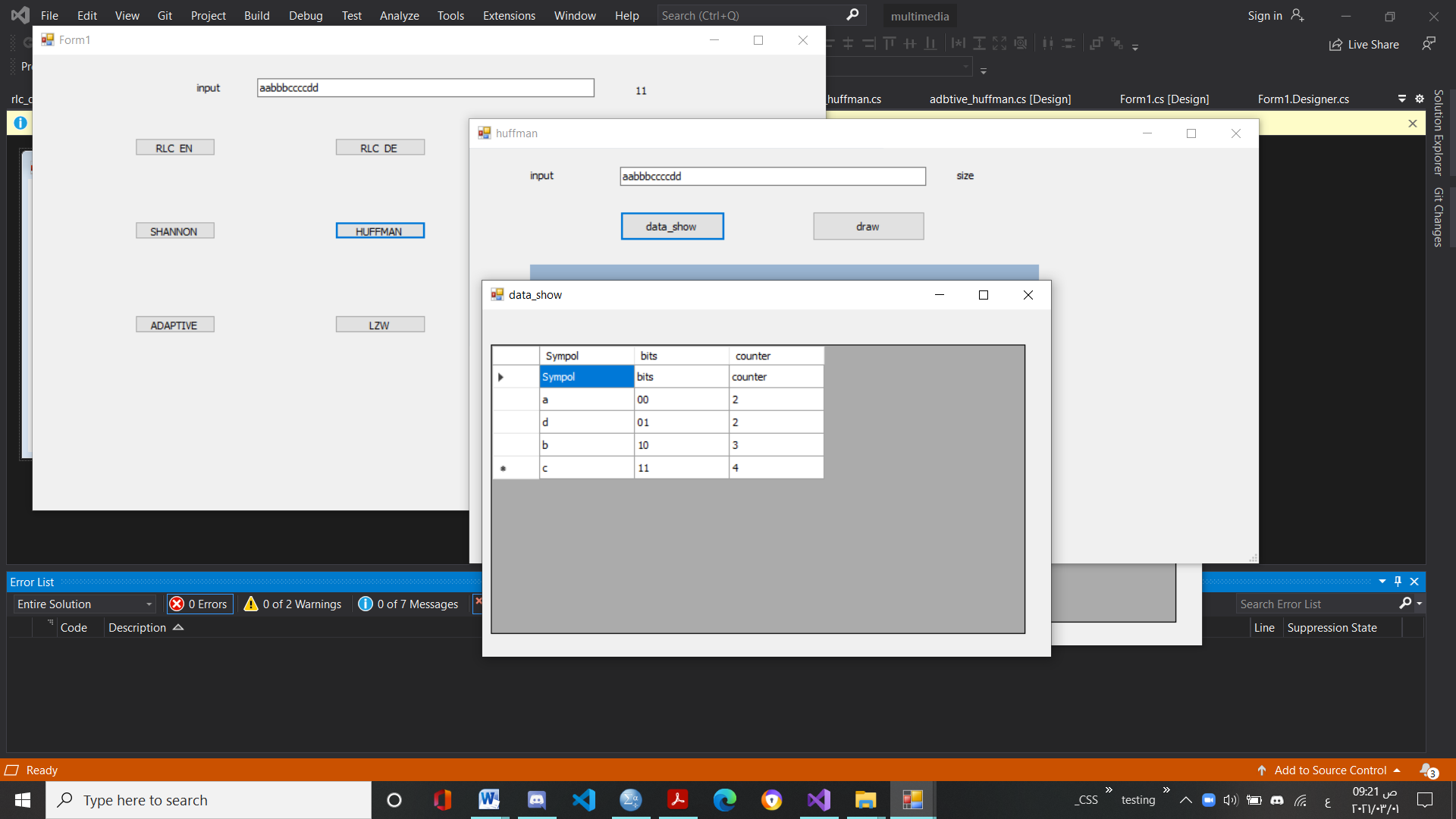
**Shannon-Fano Algorithm :-**

- Sort symbols according to the frequency of occurrences.

- Recursively divide symbols into two parts, each with

approximately same counts, until all parts contain only 

one symbol.



Huffman Coding

- Initialization: Put all symbols on a list sorted according

to frequency- Repeat until the list has only one symbol left:

o From the list pick two symbols with the lowest frequency counts. Form a Human subtree that has these two symbols as child nodes and create a parent

node.

o Assign the sum of the children's frequency counts to

the parent and insert it into the list such that the order

is maintained.

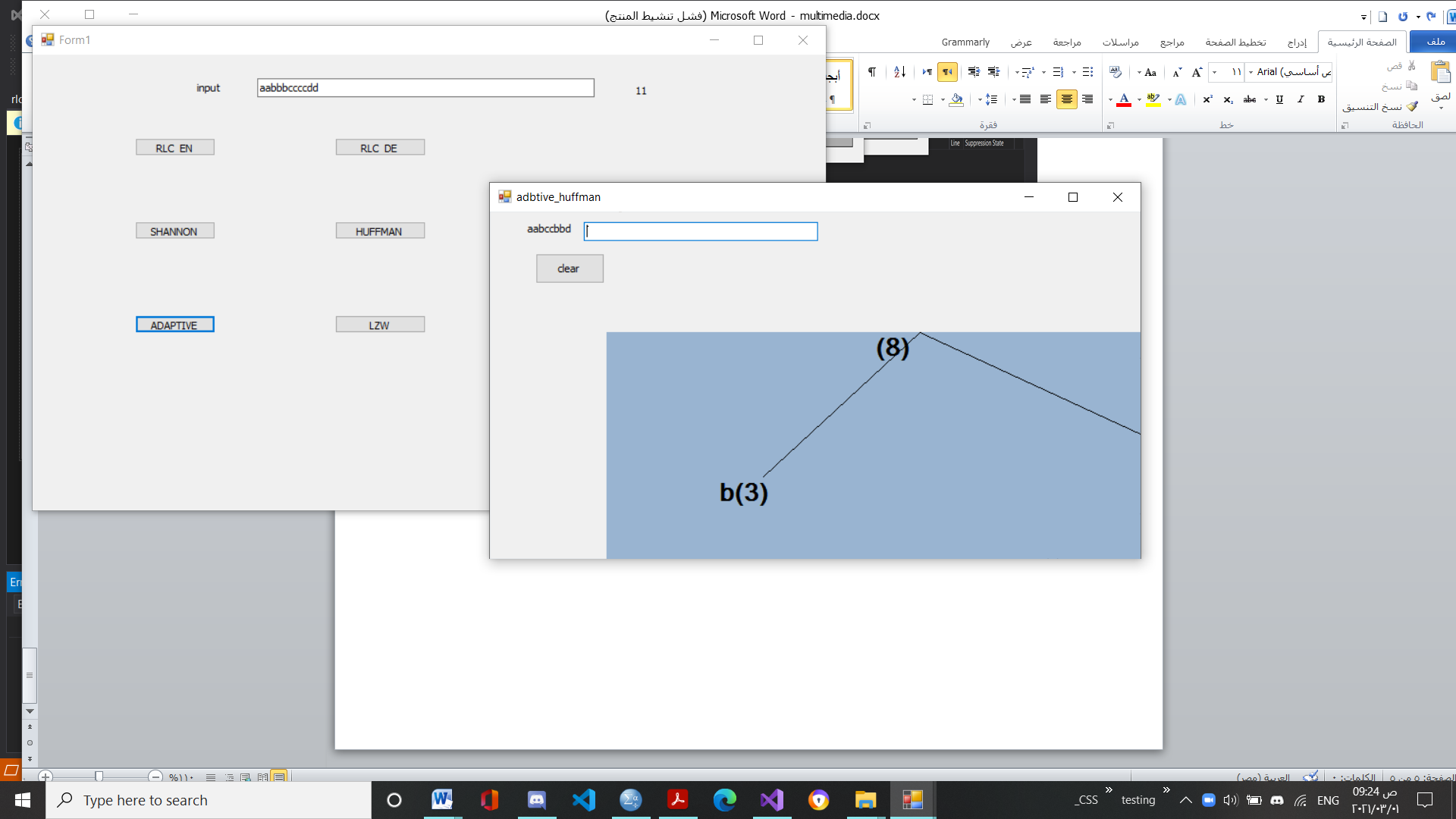
o Delete the children from the list

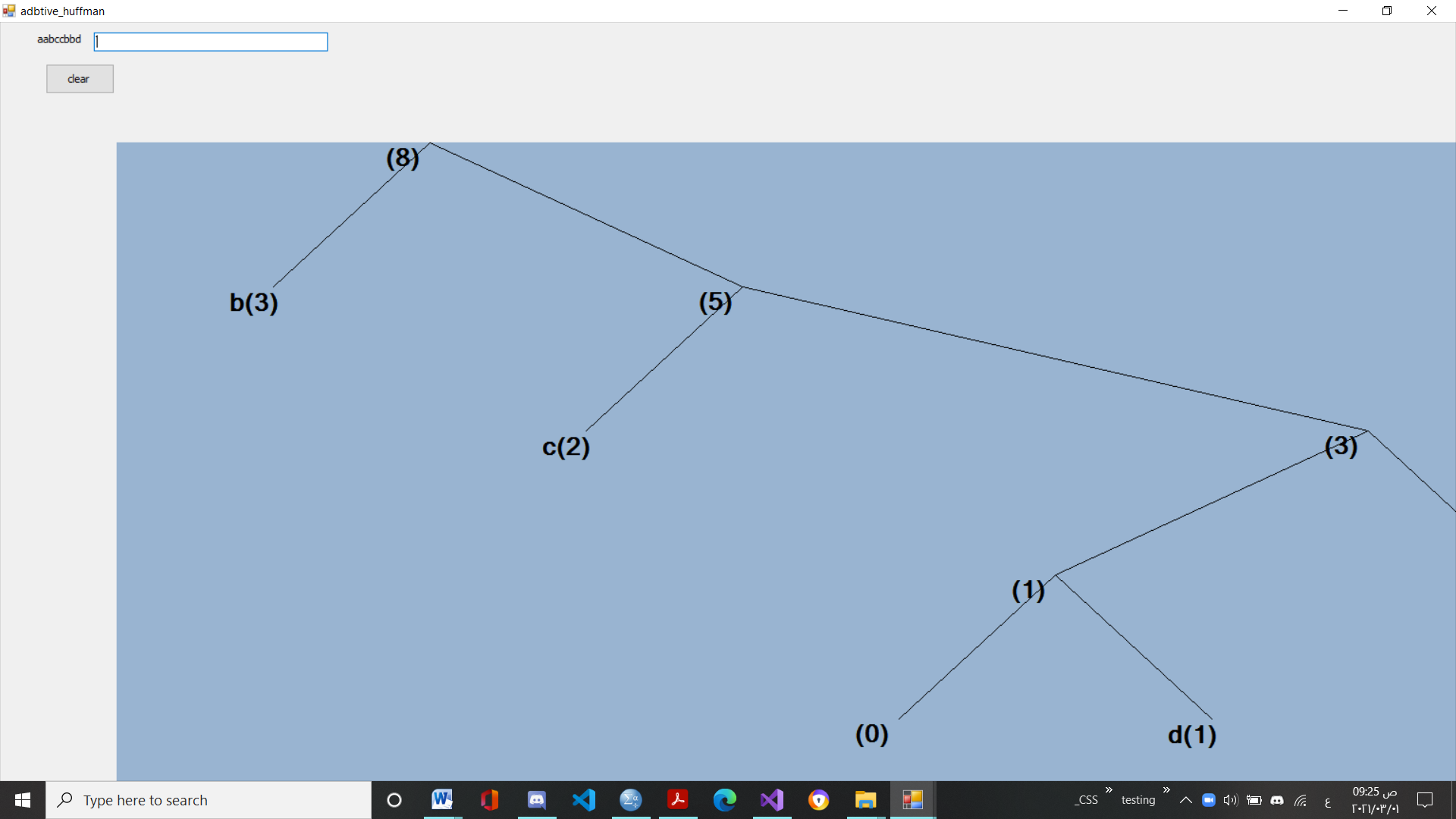
- Assign a codeword for each leaf based on the path from

the root.

- statistics are gathered and updated dynamically as data

increments the frequency counts for the symbols.





Adaptive huffman

Implicit Numbering : It simply means that nodes are numbered in increasing order by level and from left to right. i.e. nodes at bottom level will have low implicit number as compared to upper level nodes and nodes on same level are numbered in increasing order from left to right.

Invariant : For each weight w, all leaves of weight w precede all internal nodes having weight w.

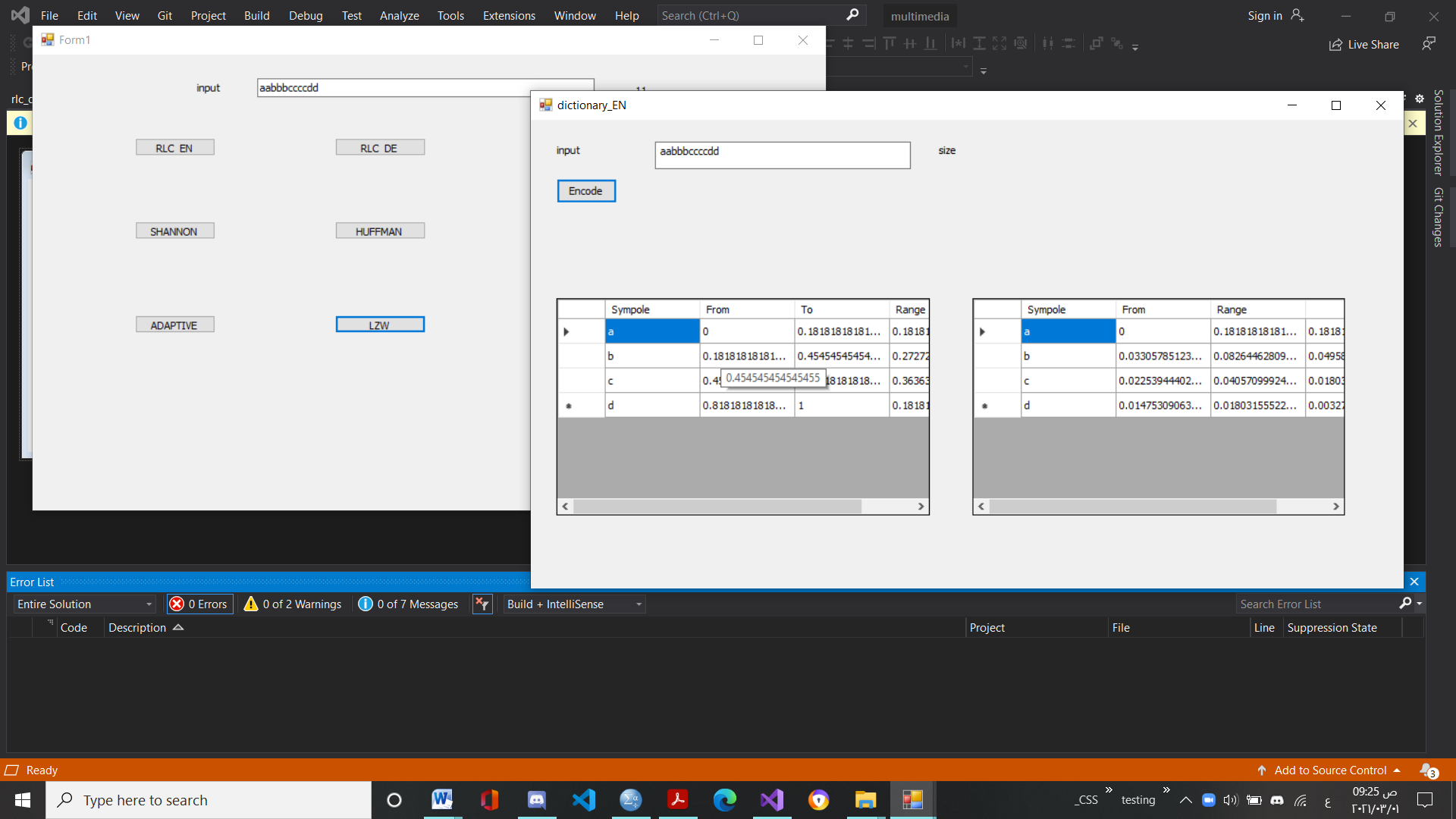
Blocks : Nodes of same weight and same type (i.e. either leaf node or internal node) form a Block.

Leader : Highest numbered node in a block.

Blocks are interlinked by increasing order of their weights.

A leaf block always precedes internal block of same weight, thus maintaining the invariant.

NYT (Not Yet Transferred) is special node and used to represents symbols which are 'not yet transferred'.



**The LZW Compression Algorithm**

can be summarised as follows:

w = NIL;

while ( read a character k )

{

if wk exists in the dictionary w = wk;

else {

add wk to the dictionary;

output the code for w; w = k;

}

}