Homework 3 - String Encoding
Due: Monday, November 7th, 11:59 PM

Congratulations Agent, you've been recruited to the NSA. Ever since <u>crackers broke into the State Department</u> files we've needed to come up with a new means to keep their prying eyes off our secrets. (Note: the opinions in this document do not reflect the opinions of me or the staff or necessarily the US Government.) The NSA already has great encryption software (built by last year's students) but it only works on numbers not characters! Stupid math… Your task, and you have to accept it, it to encode (not encrypt) a string into it's number equivalent.

ASCII Encoding

You have already learned about ASCII in class, now you just have to put it into use. For each character in the input, look it up in the ASCII table and write the decimal version of the number to the output.

`A`   becomes `065`
`AB` becomes `065066`
`0`   becomes `048`

and so on….

ASCII Decoding

Now that you can encode the message the NSA has encrypted and sent it off. They have a reply now that you must decode by doing the opposite of your encoding step.

First break the input string into usable-sized parts. (Hint, why did we write A as 065 instead of just 65?) Then simply look up the number in the ASCII table and get the corresponding character.

User Interface + Details

For this homework you will have to design your own user interface. However it must follow the following specifications.

- Use two large multiline textboxes (click on the textbox then on the small white arrow on the top right, then check "multiline"). On the left of the UI will be the the the input, on the right the output in a *read-only* textbox. You should use the Text_Changed event, not a button click to trigger the encoding. I.E. it should encode live as you type. I would start by first making whatever you type show in the second textbox as you type it.

- Use two radio buttons, one for encoding mode (characters to numbers), and one for decoding mode (numbers to characters).

- You do not need any sort of restrictive input validation. You should accept any input that the user gives.

- Note that ASCII is only defined for 0 through 255. If you want to support Unicode (for other alphabets which would be cool, you can use `AscW(Character)` and `ChrW(Character)` which handles 0 through 65535 (you will need to pad your encodings with more zeros though!)

- If decoding is not possible (for example you cannot decode "Hello" because it is not a number) you should output "~" (called a tilde, it's next to the 1 key on US keyboards). So "065" would output "A" but "0650" would output "A~" (see examples below). Do not use any error popups.

- You should not use any sort of ready made encoding decoding system. For example do not try this https://msdn.microsoft.com/en-us/library/system.text.encoding.ascii(v=vs.110).aspx. It will not do what we are trying to do.

- Some ready made functions you should use are those that we learned in class. You will probably need `Asc(Character), Chr(Integer), PadLeft(Integer, Character)` which can accept a character as it's second input (defaults to a space), and `Substring(Integer, Integer)`.

- You might also find `Math.Min(Integer, Integer)` to be useful.

Example Encodings

Characters	Encoded Numbers
A	065
AB	065066
065	048054053

Example Decodings

Characters	Encoded Numbers
065	A
0650	A~
06506	A~
065066	AB