**task - ATBASH**

description:

1. you are responsible for describing enemy communications.
2. ATBASH = A=Z, B=Y, C=X…
3. We know that the enemy is using a ATBASH cipher - decrypt the strings and then sending them.
4. you and your team need to write code that will decrypt the messages, and show how severe they are.
5. you will work in groups of 2/3: repo manager and cipher master.
6. github setup:  
   1. repo manager:
      1. create a new repo and connect it to visual studio (you can do it with the GUI)
      2. add the cipher master as a contributor to the repo.
   2. cipher master:
      1. accept the invitation
      2. create a new branch and add your name and the repo master name to the README
7. first PR(Pull Request):  
   1. now we will open our first PR on github. The cipher master finished working on the readme, and wants to merge it with main.
   2. In real life we merge by opening Pull Request (PR) in github.
   3. cipher master:  
      1. click on “new” in the PR section.
      2. go over the changes and make sure that everything there is intended to be there.
      3. open the PR.
   4. repo master:  
      1. review the PR
      2. if you approve - approve it (this will merge it behind the scenes)
      3. delete the branch
   5. both:
      1. checkout to main locally
      2. git pull - see that the main is updated and the readme has the latest changes.
      3. delete the reference to the branch in the cloud by running:   
          git fetch - -prune
      4. delete the local branch by running:  
         git branch -d <your branch name>
      5. run: “git branch” afterwards to make sure that it was deleted and only “main”/”master” remains.
8. the cipher:  
   1. now split the tasks between both of you:  
      1. write a function that takes the string and decrypt it. return the decrypted message.
      2. write a function that takes the decrypted message and looks for dangerous words in it (as a list of strings). for each word appearing in the message - add 1 point.  
           
         for example:   
         message: “our bomb will explode”   
         words: [“bomb”,”explode”]  
         points: 2  
           
         return the message and the number of total points
      3. write a function that takes the message and the total points and prints a warning based on the severity:  
         1. 1-5 points: add “WARNING” to the final message
         2. 6-10 points: add “DANGER!” to the final message
         3. 11-15 points: add “ULTRA ALERT!” to the final message.
         4. add to the final message the decrypted message and the warning + total points.
         5. print the final message
   2. for each task - create a branch and word in parallel
   3. open PR and review your partner's code when you are done with a task.
   4. fix merge conflicts if there are any. if there are, from your local machine:  
      by on the branch you want to merge (like “reverse-function”).  
      run: “git pull origin main/master”. this will pull all the commits of main in the cloud into your local “reverse-function” branch.   
      this will create merge conflicts on your local “reverse-function” branch.  
      fix them   
      commit the changes   
      push to “reverse-function” in the cloud (just “git push”) - PR should be clean
   5. test your software by running:  
        
      message:  
        
      Lfi ulixvh ziv kivkzirmt uli z nzqli zggzxp lm gsv Arlmrhg vmvnb.

Gsv ilxpvg fmrgh ziv ivzwb zmw dzrgrmt uli gsv hrtmzo.

Ylnyh szev yvvm kozxvw mvzi pvb olxzgrlmh.

Mfpsyz urtsgvih ziv hgzmwrmt yb uli tilfmw rmurogizgrlm.

Gsv zggzxp droo yv hfwwvm zmw hgilmt -- gsvb dlm’g hvv rg xlnrmt.

Dv nfhg hgzb srwwvm zmw pvvk gsv kozm hvxivg fmgro gsv ozhg nlnvmg.

Erxglib rh mvzi. Hgzb ivzwb.

words:  
bomb, nukhba, fighter, rocket, secret