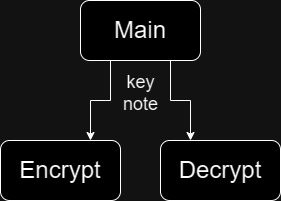
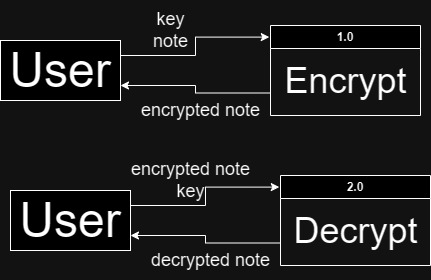
structure chart



data flow chart



pseudocode for most important function

def encrypt(int key, string note)

encrypted = ""

for character in note

if character is " " # if the character is a space it adds it to the encrypted string

encrypted += character

else if character is upper # if it is upper case it uses the upper case letters in the uni code

encrypted += chr((ord(character) - ord('A') + key) % 26 + ord('A'))

else if character is lower # if it is a lower case letter

encrypted += chr((ord(character) - ord('a') + key) % 26 + ord('a'))

else

encrypted += character

return encrypted

algorithmic efficiency

main is O(1) because it doesn't loop

both encrypt and decrypt is O(n) because it loops through each string

character and does its thing so longer the string the longer it

takes.

maintainability determination

for the simplicity of this program this is really maintainable but if

it were to get more complex it would require a lot more work to meet

the new complex encrypting method

cohesion and coupling

main is low cohesion and reasonably coupled

both encrypt and decrypt have high cohesion because they focus on a

specific cryption task. and they are loosely coupled

program trace

inputs are note = "Hello world 10?" ; key = 5 ; encrypt

| iteration | character | is " " | is upper | is lower | new character |

| 1 | "H" | false | false | true | "M" |

| 2 | "e" | false | false | true | "j" |

| 3 | "l" | false | false | true | "q" |

| 4 | "l" | false | false | true | "q" |

| 5 | "o" | false | false | true | "t" |

| 6 | " " | true | false | false | " " |

| 7 | "w" | false | false | true | "b" |

| 8 | "o" | false | false | true | "t" |

| 9 | "r" | false | false | true | "w" |

| 10 | "l" | false | false | true | "q" |

| 11 | "d" | false | false | true | "i" |

| 12 | " " | true | false | false | " " |

| 13 | "1" | false | false | false | "1" |

| 14 | "0" | false | false | false | "0" |

| 15 | "?" | false | false | false | "?" |

output is "Mjqqt btwqi 10?"

test cases

case # | name | inputs | output

1 | basic upper case encrypt | "HELLO" ; 3 ; encrypt | "KHOOR"

2 | basic lower case encrypt | "hello" ; 3 ; encrypt | "khoor"

3 | basic mixed case encrypt | "Hello World ; 5 ; encrypt | "Mjqqt Btwqi"

4 | numbers and punctuation | "123!@#" ; 5 ; encrypt | "123!@#"

5 | basic upper case decrypt | "KHOOR" ; 3 ; decrypt | "HELLO"

6 | basic lower case decrypt | "khoor" ; 3 ; decrypt | "hello"

7 | basic mixed case decrypt | "Mjqqt Btwqi" ; 5 ; decrypt | "Hello World"

8 | negative key encrypt | "Hello worlD 2024" ; -9 ; encrypt | "Yvccf nficW 2024"

9 | negative key decrypt | "Yvccf nficW 2024" ; -9 ; decrypt | "Hello worlD 2024"