Exercise 1: Enigma step by step. Useful graphical helper tool [not required to complete the exercise]:
https://fhcouk.files.wordpress.com/2012/05/pringlesenigma3a4.pdf
Below is the original wiring of the main 7 rotors used in German army and navy Enigmas in WW2:

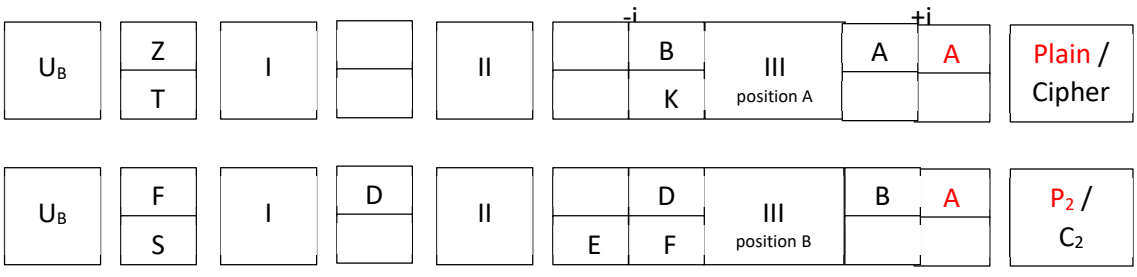
	ABCDEFGHIJKLMNOPQRSTUVWXYZ	introduced
U_B	YR Q L P GOKM B CW JAT	1937
I	EKMFLGDQVZNTOWYHXUSPAIBRCJ	1930
II	AJDKSIRUXBLHWTMCQGZNPYFVOE	1930
III	B FHJLCPRTXVZNY IWGAKMUSQO	1930
IV	ESOVPZJAYQUIRHXLNFTGKDCMWB	1938 / M3 army
V	VZBRGITYUPSDNHLXAWMJQOFECK	1938 / M3 army
VI	JPGVOUMFYQBENHZRDKASXLICTW	1939, army and navy
VII	NZJHGRCXMYSWBOUFAIVLPEKQDT	1939, army and navy
VIII	FKQHTLXOCBJSPDZRAMEWNIUYGV	1939, army and navy

Imagine that the stecker (plugboard) is empty, and we put the rotors I II and III inside the machine. So the permutation of the stecker is an identity. The next step is the so called entry wheel. In commercial Enigma [1920s] the wire from keyboard key Q would be connected to first contact A, then W to B etc. In military Enigma [too bad] it was even simpler: keyboard letter A was connected to A and so on. So the current flows from letter A at the keyboard, to entry A at the rotor III, then rotor II then rotor I and then to reflector and back. The alphabet runs clockwise when we sit in front of the keyboard and look from the right hand side of the machine, and rotors then move anti-clockwise. In addition, each rotor has ring setting which modifies the actual physical position compared to what we see in a window. In our exercise we assume that the ring setting is AAA so that if rotor is at position A, then the current enters and contact A and we obtain B with rotor III. We assume that only the rightmost (fast) rotor is moving at all. The abbreviation UB denotes the reflector also known as Umkehrwaltze B, which was changed in 1937, and which connects 13 pairs as follows:

AY BR CU DH EQ __ GL IP JX NK MO __ __, with last two pairs being omitted.

1.1. From information found elsewhere on this page, reconstruct the missing letters in the permutation U_B in the second line and in the 13 pairs above. Please use a blue pen.

1.2. We recall that the rotors move first to the next position like from A to B, and only then then the contact is made and the lamp with a ciphertext letter is on. Imagine that the machine has rotors I II II and we set up the rotors to a starting position AAZ. Then we type letter A twice. What is the ciphertext produced? Below we provide space to figure it out step by step.



Exercise 2. Incremental attack on Enigma – when plugboard settings are known.

In this exercise, we crack the commercial Enigma from 1920s used also during WW2 by Italy or Switzerland. This is equivalent to breaking the military Enigma knowing the connections of the stecker [plugboard] knowing that is some cases in WW2 the plugboard setting would be sometimes identical on the next day. Our attack is closely related variant of what was called "rodding" and was invented by Dilly Knox who has the mentor of Turing [ours presentation is very different]. We work by paper and pencil only. We continue working with rotor III at the rightmost "fast" position and we want to do a brute force search for all possible 26 positions for this rotor. We are only interested in a physical position of this fast rotor [combination of a hidden ring setting and the letter visible to the operator]. So there is only one variable i=0..25 to determine. Our goal is to deduce the exact position of this rotor. How do we verify if the offset i was guessed correctly?

*Remarks: (skip at first reading): There are two main approaches to this problem, the UNSAT and SAT method. If we knew for sure the order of rotors, and we are confident that our crib (small piece of plaintext) is accurate, then actually rejecting 25 possibilities (UNSAT) leads to an inevitable conclusion that the last choice of i is the correct one, so would not even need to check it. Now it is more interesting WHEN we can actually confirm (logically consistent or SATisfiable) a right choice with a crib. We obtain a table of Rod pairs for a permutation of 2 slow rotors and reflector (which is an involution) and the sheer amount of information in this table could already exclude any false positives (for example if we assumed that rotor II was the fast rotor.) If we get just ONE consistent solution with a crib of say 5 characters, then we do NOT have to check all the 25 possibilities for i, and we do not have to check if another rotor order might also work. Moreover there are further possibilities of **linguistic** nature (dear to Knox), to work on the crib itself. We can modify, extend or confirm the crib. For example, initially we guess that 3 consecutive letters of the plaintext are EIN. If there is no contradiction, we look at ways to extent it like K**EIN**, **EIN**SATZ, **EIN**HEIT, GEM**EIN**ER, RH**EIN** etc. In fact we need more like 5 chars start.

Below in left column $R_{III} o C^i$ we list all possible shifts by i positions of the original table of rotor III. Then we further subtract i from the result with i=0..25. The result is a composition of 3 functions C^{-i} o R_{III} o C^i, read from right to left. Finally we use this mapping to transform our **P/C pairs** (Plaintext / Ciphertext) into **Rod pairs**. These by definition are the I/O letters which would be live at the border between fast rotor and slow rotors (forward and backwards).

The goal is to see which position of the rotor III gives a consistent set of rod pairs (and/or P/C pairs). IF crib has say 5 characters or more THEN for many i=0..25, the mapping inside our rod pairs will contradict each other MANY TIMES (*it also relates together several places inside the plaintext crib). Which **line** yields a highly plausible solution i>1? Not everything need to be completed.

Plaintext / Ciphertext pairs

i	R_{III} o C^i	C^{-i} o R_{III} o C^i	EINSATZ	WUEABAI
	ABCDEFGHIJKLMNOPQRSTUVWXYZ	ABCDEFGHIJKLMNOPQRSTUVWXYZ	rod pairs left of fast rotor	
0	BDFHJLCPRTXVZNYEIWGAKMUSQO	BDFHJLCPRTXVZNYEIWGAKMUSQO	JSCJFLF	ULAEHGS
1	DFHJLCPRTXVZNYEIWGAKMUSQOB	CEGIKBOQSWUYMXDHVFZJLTRPNA		RSMFXWX
2	FHJLCPRTXVZNYEIWGAKMUSQOBD	DFHJANPRVTXLWCGUEYIKSQOMZB		
3	HJLCPRTXVZNYEIWGAKMUSQOBDF	EGIZMOQUSWKVBFTDXHJRPNLYAC		
4	JLCPRTXVZNYEIWGAKMUSQOBDFH	FHYLNPTRVJUAESCWGIQOMKXZBD	NIUHJWJ	
5	LCPRTXVZNYEIWGAKMUSQOBDFHJ	GXKMOSQUITZDRBVFHPNLJWYACE		
6	CPRTXVZNYEIWGAKMUSQOBDFHJL	WJLNRPTHSYCQAUEGOMKIVXZBDF		ZWRKLKY
7	PRTXVZNYEIWGAKMUSQOBDFHJLC	IKMQOSGRXBPZTDFNLJHUWYACEV		AXENONZ
8	RTXVZNYEIWGAKMUSQOBDFHJLCP	JLPNRFQWAOYSCEMKIGTVXZBDUH		BYOKBAG
9	TXVZNYEIWGAKMUSQOBDFHJLCPR	KOMQEPVZNXRBDLJHFSUWYACTGI		CZTNLKD
10	XVZNYEIWGAKMUSQOBDFHJLCPRT	NLPDOUYMWQACKIGERTVXZBSFHJ		

Solution can be checked with an Enigma simulator which does NOT help at all however to solve the exercise. https://cryptii.com/pipes/enigma-machine (erase all stecker pairs and set ring setting at AAA). Not so easy except with a hint, like if we can guess the positions of 2 internal rotors.

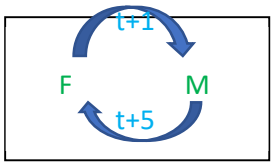
Exercise 3: Cryptanalysis with loops or early Known Plaintext Attack by Turing (which did not work well).

A bombe is basically a brute force engine for 3 Enigma rotors connected one after the other, implementing Enigma perfectly back and forth. This is implemented in many copies, or instances, all simulators run synchronously, for example one can be few steps ahead. In our example we simulate Enigma at t, t+1 and t+5 simultaneously. These multiple Enigmas a.k.a. Letchworth Enigmas will be connected together in clever ways, see below.

To avoid redoing our attack 26^3 times, which a bombe does in 20 minutes, we do just three fixed settings say ZZA, and just with rotors I II and III as before. With our online Enigma simulator we could implement the full brute force attack, from say ZZA to ZZZ: https://cryptii.com/pipes/enigma-machine. We recall that to obtain ZZA on Enigma emulator we put settings at ZZZ because the fast rotor moves prior to encryption of the 1st character. Overall for 3 positions ZZA, ZZB and ZZF we obtain with Enigma simulator the following mappings.
This is for full 3 rotor Enigma going back and forth, no stecker, ring setting = AAA:

I II III positions	ABCDEFGHIJKLMNOPQRSTUVWXYZ	offset
ZZA	EUMSANIL RHCFWTVKDPB JX	t
ZZB	WZPMOTIVGNLKDJE FYHAQUB	t+1
ZZF	QV JWDGPCYURKAOTS HZM	t+5

The student should complete the missing letters. Bletchley park/NMC bombe demonstrator guides explain the attack with a so called menu which is an undirected graph which connects various plaintext and ciphertext letter pairs. Each Enigma engine will then work on a mapping between one letter and another letter like F to M. In an extreme case a menu can have just ONE loop of size 2 like here on the right. In some sense each Letchworth Enigma is dedicated to checking one P/C letter pair , here F and M, related to encryptions at one or more time offsets t+i. The trouble is that in **no way** a simulated Engima alone, can confirm or not if this a letter pair (F,M) is a correct one. This is because we do NOT know the stecker connections. 49 bits of information are missing, a huge amount of secrecy not yet known.



Here, Turing would work by guessing: the stecker would maybe connect a certain letter say H to our P/C letter F on the menu. We assume that:

$$S(F)= H.$$

Then we run full Enigma at offset t+1 on input F using the table above to get (complete a missing letter)

$$E_k^{(t+1)}(H)=\_ .$$

In practice a bombe operator has a metallic switch to connect the wire H to the battery and the current flows at contact of a red cable A inside our bombe. We obtain that

$$S(\_)= V$$

Please complete the missing letter. We have obtained two stecker pairs for the price of one. This process CAN be iterated, similar as in Slide Attacks on Block ciphers and in our lecture slides. We obtain

$$E_k^{(t+5)}(V)=\_.,$$

another letter to complete. Most of the time this letter is NOT the same as H our initial randomly chosen letter added to our set or "hat" with stecker deductions. We deduce that (the correct answer is definitely NOT the letter H as above):

S(F)= _

In rare cases [Amplification Paradox in GOST cipher] this fails, the result could be the same letter H and we are stuck in a short loop. In Enigma the process typically continues almost forever. Up to all 26 wires will be electrically live in our circuit. The idea is that from one WRONG assumption we get another WRONG assumption about S(F), and so on. If all 26 are shown to be wrong, then no assumption is correct. This method is called SIMULTANEOUS SCANNING (term used Turing Prof's book , a fast process avoiding painful checking all the 26 possibilities for S(F)). A key point is that the electric current propagates very fast, Enigmas do not need to move, allowing and 26 cases to be checked in one step. Most of the time the conclusion is there is no solution, because our secret key (positions of 3 rotors like ZZD) was entirely not correct. In rare cases it is maybe correct and then [typically] 25 wires are alive not 26. All except the correct one, and this is obtained also in just one step. Bombe machine has a sort of display panel on the right to show WHICH of 26 wires is NOT alive.

This is how Turing has designed the first Bombe early on in 1939 and this first bombe named VICTORY did NOT work well. Turing simply did not see the full power of this type of machine. Moreover the first bomb did not even have simultaneous scanning, which is possible each time there are loops inside our menu. The more loops we have, the more additional deductions can be generated. Even if it had this ability, we have a speedup factor of 26 times, but fundamentally not better ability to break Enigma like rejecting more cases. In fact one can do yet a lot better.

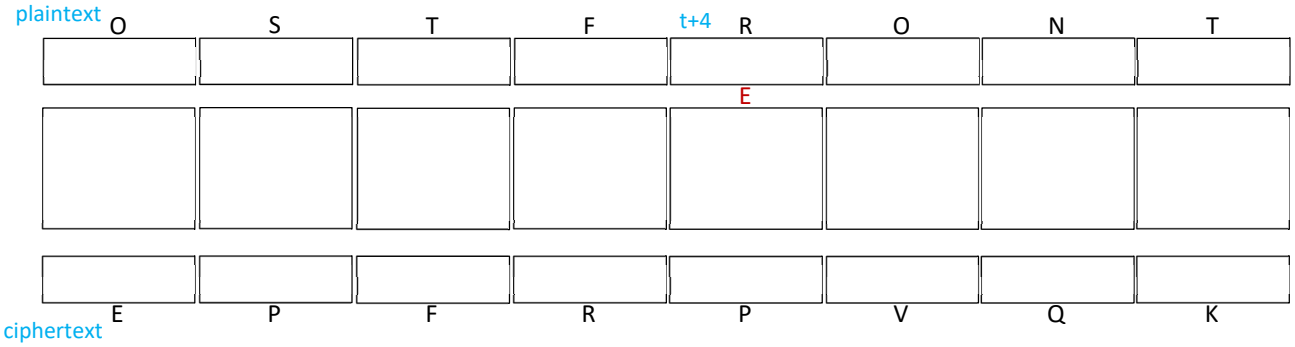
Exercise 4: Welchman attack with diagonal board (with a higher amplification ability).

Diagonal board attack was invented by Gordon Welchman at Bletchley Park. Both Turing and Welchman start by assuming one stecker pair, like maybe S(F)= H. We add one rabbit to our hat, what are the logical deductions from here? Welchman invented a method to obtain contradictions quicker, to exclude a combination of 3 rotors [not being a correct triple like ZZQ] in many more cases. In both attacks most 26^3 configurations of 3 rotors will lead to a logical contradiction. If there is no contradiction, which is a rare event, the machine will stop and the set of settings of 3 rotors will be considered to be possibly correct. First bombe had too many stops, too many false positives to check. The name diagonal board comes from how his idea was physically implemented [a lot of additional wiring added, but no additional logical components].

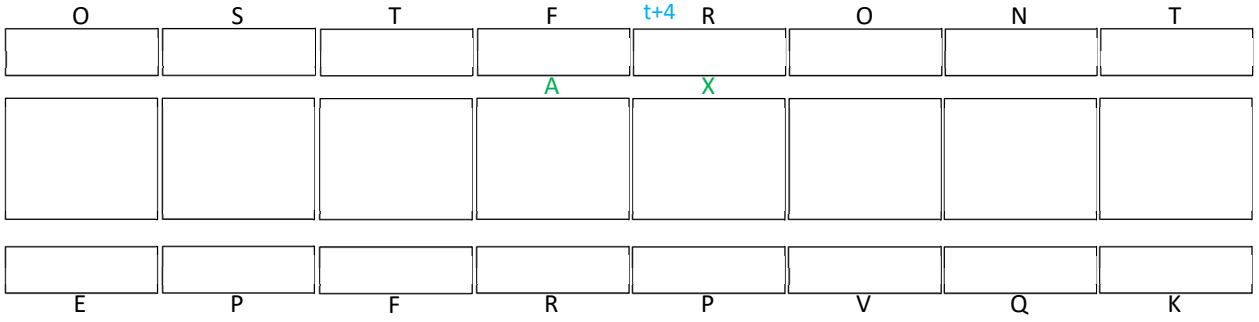
Amplification ability: While both methods can be combined, deductions using Welchman method are way more numerous [amplifying the initial guess] and making contradictions almost inevitable.

Loops and Menus. With this new method NO LOOPS in the menu are necessary. Zero loops, yet the attack almost always just works. A menu is NOT needed either. We work with the plaintext and the ciphertext directly. We are going to solve 2 examples. First we start at t=ZZA where we get a contradiction (UNSAT). If so all Enigmas on the machine advance to the next position t=ZZB where we get a situation which could be consistent (SAT = a consistent set of deductions).
Complete the puzzle:

Starting at t=ZZA up to t+7=ZZH. Here assuming S(F)= H is NOT so clever. A better way is to start with a letter being one of **O,T,P,R** all of which are repeated twice. Further it could be sensible to start with either P or R because it will provide the answer for the other. Then our assumption say S(R)= E does NOT need to be correct for our rejection to work and for the contradiction to be found. Finally with either P or R or similar it is a good idea to use one of existing letters like S(R)= E which is OK with high probability. It is important to see that the contradiction is found WITHOUT determining all stecker pairs.



Starting at t=ZZB up to t+7=ZZI, we provide some hints.



Some letters which are self-steckered are: .

We use the same table as before however we need more lines: Not all entries are needed.

I II III positions	ABCDEFGHIJKLMNOPQRSTUVWXYZ	offset
ZZA	EUMSANIL RHCFWTVKDPB JX	t
ZZB	WZPMOTIVGNLKDJE FYHAQUB	t+1
ZZC	TYOZQLINGVMFKHC	
ZZD	YREN HAT	
ZZE	XQKEDIS	t+4
ZZF	QV JWDGPCYURKAOTS HZM	t+5
ZZG	JMHI	
ZZH	CNAMFESXURZODBLQPJGVITYHWK	
ZZI	PKR	t+8