Problem 1.

Java code converts English plaintext into continuous sequence of Natural decimal integers, where: “a”=0, “b”=1 … “z”=25, “space”=26.

Then Java code divides a resulted sequence into blocks of 25 integers starting from left. Last block is *padded* with 0’s to obtain 25 symbols too in it.

I have reconstructed the indicated “3 Rounds” hash process to simplify and speed up hash calculation. In fact, I have used the following algebraic (arithmetic) transformations as “one Round” to REPLACE all these 3 Rounds. Also “mod 27” process is performed one time with last block output.

The following part here is the detailed explanation of my “one Round” process:

At Round 1 indicated in Problem1 the following summing is performed in fact for block # ***i***:

M1 = N1 + N6 + N11 + N16 + N21
M2 = N2 + N7 + N12 + N17 + N22
M3 = N3 + N8 + N13 + N18 + N23
M4 = N4 + N9 + N14 + N19 + N24
M5 = N5 + N10 + N15 + N20 + N25
where N1, N2…N25 are Natural integers with serial numbers 1,2…25 into block ***i***.

Note: 1) Output of previous block # ***i-1*** will be added **after Round 3 replacement.**
2) Subtracting of “mod 27” will perform **after Round 3 replacement.**

At Round 2 indicated in Problem1 the following summing is performed in fact for block # ***i***:

P1 = N2 + N8 + N14 + N20 + N21
P2 = N3 + N9 + N15 + N16 + N22
P3 = N4 + N10 + N11 + N17 + N23
P4 = N5 + N6 + N12 + N18 + N24
P5 = N1 + N7 + N13 + N19 + N25
where N1, N2…N25 are Natural integers with serial numbers 1,2…25 into block ***i***

Note: 1) Summing with sums M1, M2, M3, M4, M5 from Round 1 will be **after Round 3 replacement**.
2) Subtracting of “mod 27” will perform **after Round 3 replacement.**

At Round 3 indicated in Problem1 the following summing is performed in fact for block # ***i***:

R1 = N21 + N16 + N11 + N6 + N1
R2 = N2 + N22 + N17 + N12 + N7
R3 = N8 + N3 + N23 + N18 + N13
R4 = N14 + N9 + N4 + N24 + N19
R5 = N20 + N15 + N10 + N5 + N25
where N1, N2…N25 are Natural integers with serial numbers 1,2…25 into block ***i***

Note: 1) Summing with P1, P2, P3, P4, P5 will be **after Round 3 replacement**
2) Subtracting of “mod 27” will perform **after Round 3 replacement.**

Therefore, at my “one Round” process to obtain the output S1(*i*), S2(*i*), S3(*i*), S4(*i*), S5(*i*) of block *i* the following summing is performed:

S1(*i*) = S1(***i-1***) + [M1(*i*) + P1(*i*) + R1(*i*)] = S1(***i-1***) + [2*N1 + 2*N6 + 2*N11 + 2*N16 + 3*N21 + N2 + N8 + N14 + N20]

S2(*i*) = S2(***i-1***) + [M2(*i*) + P2(*i*) + R2(*i*)] = S2(***i-1***) + [2*N2 + 2*N7 + 3*N22 + 2*N17 + 2*N12 + N3 + N9 + N15 + N16]

S3(*i*) = S3(***i-1***) + [M3(*i*) + P3(*i*) + R3(*i*)] = S3(***i-1***) + [2*N3 + 2*N8 + 2*N13 + 2*N18 + 3*N23 + N4 + N10 + N11 + N17]

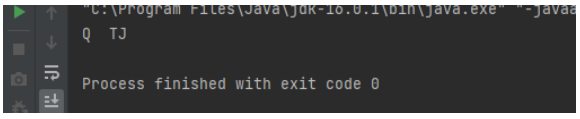
S4(*i*) = S4(***i-1***) + [M4(*i*) + P4(*i*) + R4(*i*)] = S4(***i-1***) + [2*N4 + 2*N9 + 2*N14 + 2*N19 + 3*N24 + N5 + N6 + N12 + N18]

S5(*i*) = S5(***i-1***) + [M5(*i*) + P5(*i*) + R5(*i*)] = S5(***i-1***) + [2*N5 + 2*N10 + 2*N15 + 2*N20 + 3*N25 + N1 + N7 + N13 + N19]

Note: For S1(***1***), S2(***1***), S3(***1***), S4(***1***,), S5(***1***) of 1^{st} block S1(***1-1***)=S2(***1-1***)+S3(***1-1***)=S4(***1-1***)=S5(***1-1***) **= 0**

Then with output S1(*i*), S2(*i*), S3(*i*), S4(*i*), S5(*i*) **of last block** “mod 27” process is performed .
And finally, the return conversion into English letters and “space” symbols is performed to obtain Hash Value.

INPUT: abcdefghi jklmnopqrstuvwx



Hash value for the plaintext: “the birthday attack can be performed for any hash functions including sha three” is UDOIY

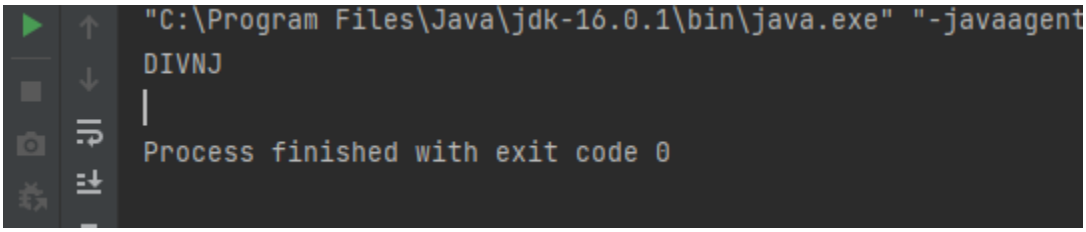
Problem 2. #INPUT SHOULD BE LOWER CASE LETTERS

I have selected the key **K = “kplatzblxmoeziy”,** therefore, **K II x** = “**kplatzblxmoeziythe birthday attack can be performed for any hash functions including sha three**”

After the hash function the calculated Hash value **h(K II x)** = **DIVNJ**

Correct Phrase for the DIVNJ output

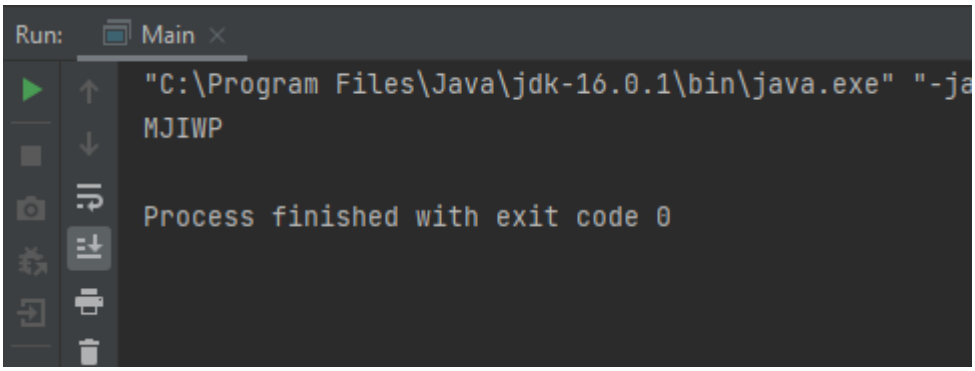
kplatzblxmoeziythe birthday attack can be performed for any hash functions including sha three



Then I have added key **K** again to the calculated Hash value: **K ll h(K ll x)** = "**kplatzblxmoeziydivnj**

And I have calculated Hash value again: **h(K ll h(K ll x)** = **MJIWP**

So, MAC_K (x) = MJIWP



Problem 3.

I have tried "birthday" attack to hash function in Problem 1. Some not significant signs, namely, additional "spaces" were added to both correct and fault. messages**: "the birthday attack can be performed for any hash functions including sha three**" and "**the birthday attack can not be performed for any hash functions including sha three**". Hash value results these messages are shown in Table.

Correct message variations	Hash value	Fault message variations	Hash value
the birthday attack can be performed for any hash functions including sha three	UDOIY	the birthday attack can not be performed for any hash functions including sha three	PNIEC
the birthday attack can be performed for any hash functions including sha three	BZXIJ	the birthday attack can not be performed for any hash functions including sha three	AZYWW
the birthday attack can be performed for any hash functions including sha three	LZYRQ	the birthday attack can not be performed for any hash functions including sha three	WQPBM
the birthday attack can be performed for any hash functions including sha three	SRFBZ	the birthday attack can not be performed for any hash functions including sha three	HJANK
the birthday attack can be performed for any hash functions including sha three	RZYNO	the birthday attack can not be performed for any hash functions including sha three	SRP R
the birthday attack can be performed for any hash functions including sha three	BGJKN	the birthday attack can not be performed for any hash functions including sha three	YERBU
the birthday attack can be performed for any hash functions including sha three	UDOHW	the birthday attack can not be performed for any hash functions including sha three	ATNNV

Hash value has 25 bits (5 symbols, each of 5 bits), therefore, for a probability of successful attack (to obtain collision) as ~0.5 the number of variations should be as $2^{25/2}$ = 5,793.

But I have done **14 variations** only, and collision was NOT found, due to a probability of successful attack is very low in my case with 14 variations.