Assignment- 1

Harshit Rai
2017152

Question- 1

Ans. 1)

Total special characters = 33
Total digits in base 12 = 12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B)
Total alphabets = 52 - 2 = 50
↓
(Upper-case, lower-case, removing A, B)

As we do not have any special character like alpha and beta to represent the numbers 10, 11. We will take 2 alphabets to represent them. (A and B as 10, 11).

Total passwords which start with an alphabet = 50 × 95 × 95 × 95 × 95 × 95
— → Alphabet
— — — — — → Anything (33 + 12 + 50)

Total choices = $50 \times (95)^5 = 3.868904688 \times 10^{11}$ (A)

Now among the total choice, we have 4 different cases:

(B) Passwords having at least one digit and no special character.

(C) Passwords having at least one special character and no digit.

(D) Passwords having both digits and special characters.

(E) Passwords having only alphabets in it.

$A = B + C + D + E$

$D = A - (B + C + E)$ — (I)

↓

Total Passwords starting with an alphabet and having both digits and special characters.

(B)

Total Passwords having at least one digit and no special character. $= (52) \times \underline{62} \times \underline{62} \times \underline{62} \times \underline{62} \times \underline{62}$

↓ Alphabet

$\times \underline{62}$ → (Alphabet or digit, 50+12)

$= (50) \times (62)^5$

But this will also contain cases when all of them are alphabets and no digit. So we have to subtract it from this.

$(B) = (50) \times (62)^5 - (50)^6$

↓ only alphabets

$(B) = 3.01816416 \times 10^{10}$

(C)//

Total Passwords having at least one special character and no digit in it. $= (50) \times (83)^5$

((50): Alphabet; $(83)^5$: Alphabet or special character, $50+33$)

But this will also contain cases when all of them are alphabets and no special character. So we have to subtract it from them.

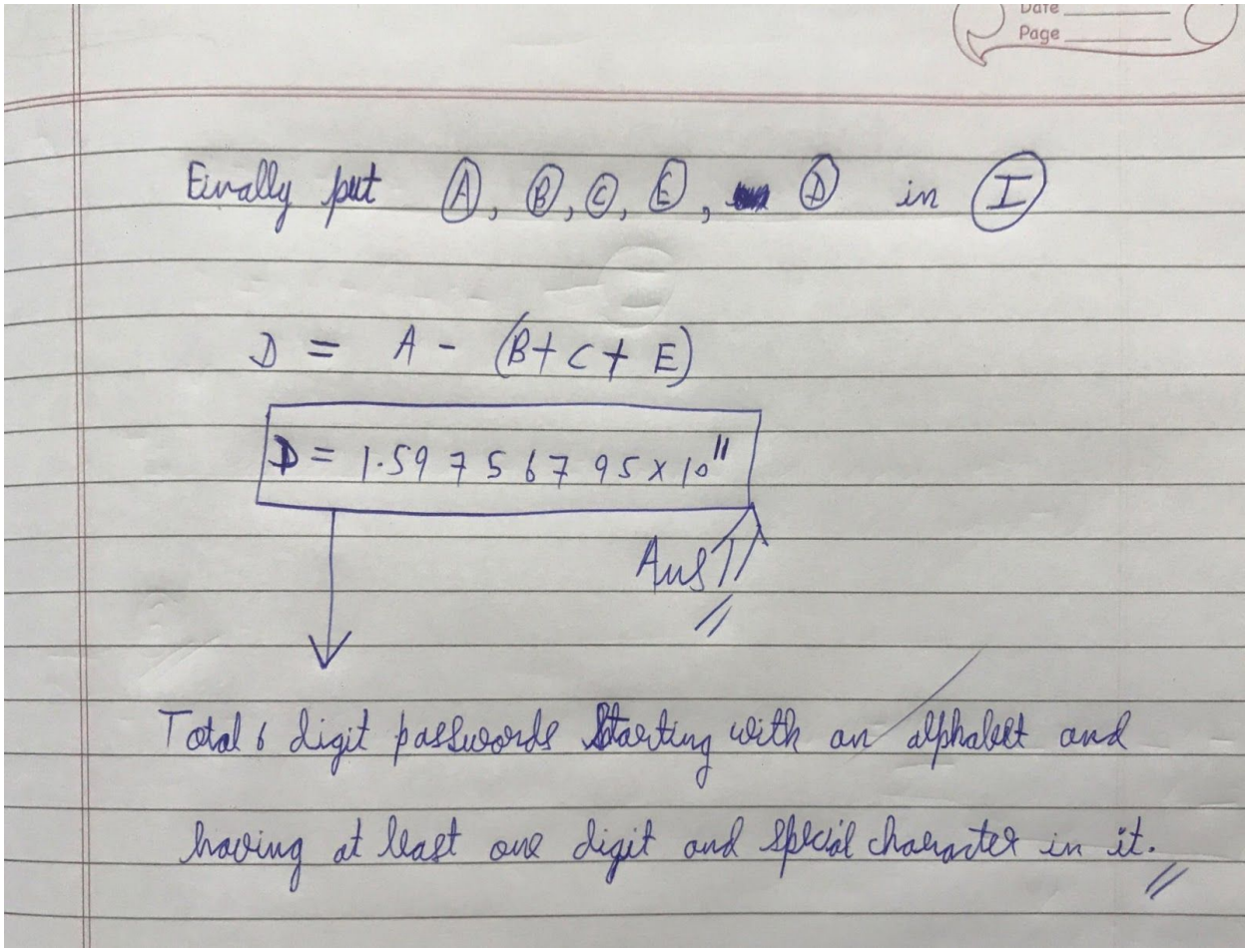
$$(C) = (50) \times (83)^5 - \underbrace{(50)^6}_{\text{only Alphabet}}$$

$$(C) = 1.8132703222 \times 10^{11}$$

(E)//

Total passwords having only alphabets in it. $= (50)^6$

$$(C) = 1.5625 \times 10^{10}$$



Question- 3
Password Policy:

- Minimum 8 characters. (1 point and reducing 1 point if it does not satisfy)
- It should not be a dictionary word. (1 point and reducing 1 point if it is)
- Should contain at least one uppercase character. (1 point and reducing 1 point if it does not have any)
- Should contain at least one lowercase character. (1 point and reducing 1 point if it does not have any)
- Should contain at least one special character. (1 point for each special character and reducing 1 point if it does not have any)
- It should contain at least one digit. (1 point and reducing 1 point if it does not have any)
- Total distinct characters should be strictly more than 5. (1 point and reducing 1 point if it does not satisfy)
- The maximum frequency of any character should not exceed 3. (1 point and reducing 1 point if it exceeds)

1. q2w3e4r5t - **Neutral** (4 Points)
2. Football - **Weak** (2 Points)
3. 983749587 - **Weak** (2 Points)
4. 4><8mM% - **Very Strong** (8 Points)
5. FCSPassword1 - **Strong** (6 Points)
6. &*^)&@_(%$ - **Very Strong** (11 Points)
7. SecurePassword@123 - **Very Strong** (8 Points)
8. monkey - **Very Weak** (-2 Points)
9. Hello! - **Weak** (2 Points)
10. Qwertyuiop - **Neutral** (4 Points)

Very strong: **6< Points**
Strong: **5 or 6 Points**
Neutral: **4 Points**
Weak: **2 or 3 Points**
Very weak: **2> Points**

Question- 4

A) For designing a robust online voting system, I would take majorly Confidentiality, Integrity, Availability, and Usability into consideration. Confidentiality and Integrity have been explained in “Part C” of this question. The voting system should be consistent, the total submitted votes from the user end should be equal to the total number of votes in the database. The system should be accurate, the choices made by the user should get reflected in the database correctly. Also, the system should be reliable.

The system should be designed in such a way that in case of any mishappening, the data must not be completely lost. There should always be a backup and for that purpose, one can store the data on the cloud platform as well. Also, data retrieval should be fast and the system must not crash. Usability is also a major concern, the system must not be too much complicated that the users are unable to use/understand. We can implement a fingerprint scanner while voting for maintaining the uniqueness of the user. Here the security is increased but on the other hand, the usability is somewhat decreased.

B)
A= { string of exactly 7 digits }. Roll No. is unique to every student and is linked to every IIIT-D student.

C= { unique hash ID }. It is the information stored in the database which is used to validate authentication information. This will ensure that no outsider logs into the system.

F= { strong hash functions }. Complementation function, it changes **"A"** to **"C"**. Where **"A"** is a known quantity and **"C"** is an unknown unique hash ID.

L= { Login }. Functions that prove identity.

S= { Password }. Only after entering the right password, the user will be allowed to submit his/her vote into the database otherwise no change is done to the database. This will ensure that no user logs into another user's profile.

C) Confidentiality:

- Eligibility: I will make it compulsory to log in using their IIIT-D ID's into the system before voting, this will ensure that no outsider is voting.
- Uniqueness: Maintaining a flag (initially false) for each user which becomes true after the user is done with his voting. This will ensure that one person gets to do at a max of 1 vote.
- I will encourage everyone to make good password choices. The password should contain a minimum of 8 characters, an upper-case, a lower-case, a digit, a special character and it must not be a dictionary word.
- I will enable data encryption and two-factor authentication in my system. This will ensure that no one can decode whom the other person has voted for except the system itself.

Integrity:

- Using a strong Hash function for encryption. So that all votes are collected and stored in a safe database.
- The database should be "write-only-once" mode for the users so that no one alters the database.
- The database should be"read-only" mode for the administrator. So that others can not access the database except the administrator.
- Enabling Audit trail and timestamping will prevent the database from improper or unauthorized changes.

Question- 5

Before actually breaking into the bank, I will try to get some preliminary information about the bank. For that, I will use the **'Pretexting'** technique by calling the bank several times and impersonating someone who has the right to know the following details:

1. How many tellers work there, details of the manager: **(Tax-official)** I'm speaking from <XYZ> tax-office, there is a slight confusion in your last month's income tax bill. It would

be great if you could answer my queries on call, like how many employees work there, the gender ratio, and a copy of your last months' income tax bill.

2. How many security cameras: **(Electrician)** I'm speaking from <XYZ> electrical, due to some technical difficulties, there might not be electricity on <Planned date>. I would like to know whether you guys have a generator or not. Please tell me about the number of PCs, cameras, and other heavy equipment as we have to calculate the total electricity load.
3. What is lunchtime: **(Plumber)** I'm speaking from <XYZ> sewage company. Due to some new government policies, your building's pipeline system needs to be reviewed as we have detected a fault in your pipeline system. It would be bothersome if we came in your work hours. So, could you please tell us at what time do you guys have your lunch? And we can come at that time.
4. Map layout of the bank: **(Building maintenance or a Carpenter)** I'm speaking from <XYZ> company. As an order from the government, we have to inspect the buildings in your area ASAP as it might cause you trouble if the police come first and see that your facility has not been reviewed yet. Can you please provide us with your building's blueprint for the same? I'm providing you my e-mail address (A fake email address with real name).

While I'm on the call, I will use the '**Borrowing the HOLD music'** technique, in which I will intentionally put the call on hold and play the original company's HOLD music, just to sound more authentic that the call is indeed from the company itself and not a hoax call. Also, after getting the name of the manager, I can quickly look after his family and personal details.

Now I will make a fake government order stating that the bank's pipeline system is not according to the new policy. Also, I have to convince a real plumber by giving him a small **bribe** to help me out with this inspection. His ID card and other details will be authentic, unlike mine. So I will use him as a **'piggyback'** to access the bank.

Now I have some information about the bank. The next day, my colleague and I will go to the bank just 15-20 minutes before lunchtime (**to avoid the crowd**) dressed as a plumber as it is our right to inspect the building. Also we have the orders from the government (fake) and a genuine ID card (My colleague). To avoid any suspicions, I will act in a very **naive manner**. I will intentionally make **small mistakes** like forgetting the bank's name or my company's name when asked, stuttering on the sentences. I will act **nervously**, unlike my colleague. This will **lower their guards** as they will start to think that I can't harm them; only my partner is good at his job. I will go to that counter, which has an old lady or female, preferably as the teller because females are **very emotional and fearful** compared to men. After finally finding the perfect teller, I will compliment her to **build trust,** asking her that my partner and I are here to inspect the building's pipeline system, can I please talk to the manager.

When the manager arrives, I will ask him to please cooperate with us. I will use my partner's ID card to **'tailgate'** the manager's security access after inspecting all the other rooms when we

arrive at the vault. I will ask my colleague to wait at the outside door and look for the pipeline according to the blueprint he has (But he will go and have a chit chat with the security guards just to lower the chances of being caught). As soon as the manager and I go into the vault, I will use the **'Vishing'** technique. I will quickly pull out my toy gun from my bag, which I bought in by **'piggybacking'** the security guard, and will threaten him to open the vault. I will make him feel **helpless and frightened** by **targeting** his family by saying that if he tries to call the police or shout, my colleague will hear you and lock both of us in this room, which might even cost you your life. He might even harm your family members. If he disagrees with this, I will try to give him a '**bait'** by saying, you can keep some money as well; no one is going to know about it. **Is money more important or your life?** After all, this has happened, it will create a feeling of **fear, greed, and urgency** in his mind, and he should open the vault by now. If he opens the vault, I will make sure that there are no **dye packs or panic buttons** between the dollar bills. If my plan fails, I would somehow just have to hit the manager to make him faint in the room and run through the shortest and fastest way possible.