Question 3

- **The difference between AI and General AI**

While Al focuses on a single, defined purpose, General AI is still in its early stages of development, with some developers doubting and researching the potential of its existence in the future. Image recognition, hyper-personalization, chatbots, and predictive text are all examples of AI. Machines that can handle a wide range of cognition tasks with minimum supervision are sought by general AI. Data scientists train AI to perform specific jobs. General AI, on the other hand, involves the ability to learn, generalise, apply information, and make future plans. The main distinction between the two is that AI lacks self-awareness, consciousness, or the ability to think, but General AI has common sense, creativity, and the ability to express emotions. To complete a task, AI must correlate problems or tasks to a specific data set, whilst General AI must continuously pass the Turing test.

- **Turing Completeness**

If a system of data manipulation rules (such as a computer's instruction set, a programming language, or a cellular automaton) can be used to simulate any Turing machine, it is said to be Turing-complete or computationally universal in computability theory (devised by English mathematician and computer scientist Alan Turing). This means that the system can recognise and decide on different data manipulation rule sets. Turing completeness is a measure of how powerful a set of data manipulation rules is. Today, almost all programming languages are Turing-complete. It is named after Alan Turing, the developer of a Turing machine and a computer scientist.

For example, particular rule systems which access and modify data include computer languages and CPU application programs. The rules are considered to be "Turing complete" if they can emulate Turing's hypothetical computing machine. Mathematically, a Turing-complete system can be shown to be able to perform any calculation or computer programme. Lambda calculus, established by Alan Turing's tutor Alonzo Church, is an example of a Turing complete system. C and Pascal are two examples of procedural programming languages. Object-oriented languages, such as Java and C++, are the most used.

- **Discuss whether Bitcoin is Turing Complete**

There is a mixed reaction by people on whether bitcoin is turing complete or not. Loops are not presently supported by Bitcoin scripts. As a result, they are often regarded as not being Turing Complete. The Bitcoin script can only execute linear or tree-like algorithms as a result of this restriction. According to me bitcoin is turing complete. Craig Wright explained why is it so. Taking the outputs of almost any Bitcoin transaction then using them as inputs in the following one. The Bitcoin Script (at least in BSV) can calculate anything and produce a proven state, whose signature can be transferred to the next block and used for something else. Despite the fact that one stage of the computation must be completed inside the block confirmation time, the result is ready for the following step. Data can be added to the

following transaction, but it will only validate if the previous information is already on-chain. An "overlay network" is what you get when you combine Script and external mechanisms in this way—and even if that doesn't fit your notion of theoretical Turing Completeness, it's irrelevant because Bitcoin can execute any computational task needed of it. (CoinGeek,2017)

- **Discuss what role blockchain might play working with AI in the future**

➢ Assist In Building a Distributed Supercomputer:

Because we can't always grasp every AI's conversation with us, AI technology struggles from a vast number of unexplained data values. AIs transmit or produce data in the form of values or streams. What AI blockchain can accomplish in this case is assist you in the creation of a distributed supercomputer. The combination of AI and blockchain as a framework would be able to improve AI's overall experience. This would result in better actions, creative results, and artificial intelligence that was smarter. Ignoring the network would allow the AI to learn more quickly than before, resulting in a far more efficient AI development process.

➢ Connect Different Marketplaces:

Blockchain will not only be able to connect all markets at once, but it will also allow new enterprises to emerge. This will kick off a new data market. The creation of any AI would likewise be decentralised if data was stored on the blockchain. As a result, any AI operation's confidentiality would be erased, and anyone may benefit from it. It will also result in the creation of an AI marketplace where you can purchase AI development tools from other organisations.

➢ Resolve The Trust Issue:

By storing everything in a decentralised database, one can see exactly whatever the AI does and how it communicates with other instances. It will restore your faith in AI, allowing you to rely on it with confidence. Interactions between machines will be recorded as well, so you'll have a greater understanding of how they communicate.

➢ Improve the AI-User Experience Overall:

The AI blockchain will provide users with better AI experience of their lives. Blockchain already prioritises customer experience and security over all else. However, when AI is powered by blockchain, the potential for fresh applications is endless.