**Q1. In Python 3.X, what are the names and functions of string object types?**

**ANS:-**

**In Python 3.X and later, String objects has bulit in class named as "str" and it is used to store characters in memory in encoded format.**

**Q2. How do the string forms in Python 3.X vary in terms of operations?**

**ANS:-**

**Python supports ASCII as a subset of Unicode. The default encoding of characters in Python is UTF-8 (Unicode Transformation Format - 8-bit).ASCII is really a 7-bit character set; it is mapped to 8-bit bytes by setting the high bit to zero. Thus, ASCII characters are unchanged in UTF-8. A non-zero high bit indicates that a character composed of multiple bytes.**

**u = 'akash'**

**print(ord(u[-1]))**

**104**

**Q3. In 3.X, how do you put non-ASCII Unicode characters in a string?**

**ANS:-**

**As we know, there are only 127 ASCII characters in ascii table. It stores symbols, letters, whitespace characters, non printable characters, which are very less. So Python 3 has adopted unicode encoding of strings. for non ASCII u can get any value more than 127.**

**s = 'Akash'**

**s += chr(128)+chr(169)**

**print(s)**

**Akash€©**

**Q4. In Python 3.X, what are the key differences between text-mode and binary-mode files?**

**ANS:-**

**Files opened in binary mode (appending 'b' to the mode argument) return contents as bytes objects without any decoding. In text mode (the default, or when 't' is appended to the mode argument), the contents of the file are returned as strings, the bytes having been first decoded using a platform-dependent encoding or using the specified encoding if given.**

**Q5. How can you interpret a Unicode text file containing text encoded in a different encoding than your platform's default?**

**ANS:-**

**While reading file, we can specify the encoding such as(utf16, utf32), so the file will be decoded according to that encoding and results are returned to the user.**

**text = "this is sample text file for understanding encoding"**

**f = open('sample.txt','w', encoding='utf16')**

**f.write(text)**

**f.close()**

**f = open('sample.txt','r', encoding='utf16')**

**print(f.read())**

**this is sample text file for understanding encoding**

**Q6. What is the best way to make a Unicode text file in a particular encoding format?**

**ANS:-**

**text = 'ହୋଉ ହେଲା ଓଡ଼ିଆ | ବହୁତ୍ ଛଇ ତୁମର'**

**encoded\_data = text.encode("utf16")**

**file = open("sample1.txt", "wb")**

**file.write(encoded\_data)**

**file = open("sample1.txt", "r", encoding='utf16')**

**print(file.read())**

**ହୋଉ ହେଲା ଓଡ଼ିଆ | ବହୁତ୍ ଛଇ ତୁମର**

**Q7. What qualifies ASCII text as a form of Unicode text?**

**ANS:-**

**The first 128 Unicode code points represent the equivalent ASCII characters. They have the same meaning in both ASCII and unicode. Since UTF-8 encodes each of these characters with a single byte, any ASCII text is also a UTF-8 text. Unicode is a superset of ASCII.**

**Q8. How much of an effect does the change in string types in Python 3.X have on your code?**

**ANS:-**

**In Python 3.X and later, there is only one string type that is unicode. So no effect on the code.**