Name: Abhishek Singh Bisht
Course: B.C.A
Sec: A
Roll No: 1121005

## Answers

(1) (a) Public key of sender and private key of recievers

(2) (c) Spyware

(3) (b) A public and private key

(4) (c) Cyber law

(5) (a) Only on alphanumeric

(6) (d) Idea is same title is different

(7) (a) hash value

(8) (c) The position of the character is changed in spite of its identity.

(9) (d) both b and c

(10) d None

Name: Abhishek Singh Bisht

Course: B.C.A

Sec: A

Roll No: A

Answer

A (2) 3 security aspects of the browser (Chrome):

(1) Task Manager

Chrome has its own Task manager that shows you how much memory and CPU usage each tab and plug-in is using. You can open it by clicking shift-Esc from within chrome. You can get more details by clicking the 'stats for nerds" link on the task manager and it will open a page with a full breakdown of memory and CPU usage for each process within the browser.

(2) One box for search, address and history

Instead of having separate input boxes for the search bar and the address bar, Google has united the two in chrome. Google also added history to the equation.

(3) Upgraded Tabs

In chrome you can drag a tab into its own window, and drag it back to the

main window. This is called "Dynamic tabs". Also, by default, the "New tabs" page in Chrome features a page that shows thumbnails of your most visited web sites, a list of your recent bookmarks, and a search box that allows you to search your history.

Name: Abhishek Singh Bisht

Course: B.C.A

Sec: A

Roll No: 1121005

## Answers

(3)
```
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) - len(key)):
            key.append(key[i % len(key)])
    return("".join(key))

def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) + ord(key[i])) %26
        x += ord('A')
        cipher_text.append(chr(x))
    return("".join(cipher_text))

def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) - ord(key[i]) + 26) %26
        x += ord('A')
```

```
        orig_text.append(chr(x))
    return("".join(orig_text))

if __name__ == "__main__":
    string = "Cryptography"
    keyword = "Monarchy"
    key = generateKey(string, keyword)
    Cipher_text = cipherText(string, key)
    print("Ciphertext:", cipher_text)
    print("Original/Decrypted text:", originalText(cipher_text,
          key))
```

Name: Abhishek Singh Bisht

Course: B.C.A

Sec: A

Roll No: 1121005

Answers

A (4)

```
import math, random
def generate_OTP():
    digits = "0123456789"
    OTP = ""
    for i in range(4):
        OTP += digits[math.floor(random.random()*10)]
    return OTP
if __name__ == "__main__":
    print("OTP of 4 digits:", generateOTP())
```