

**m nm m ;;;jkkkkkkkk jjjjjjjjjhjhoo j;on j i hjn j j bj nj jj nihuihppnn ubib bn h hjb lb liu**

Block 3



[Date]

[Company name]

H 7iou89 u87u u89

Question No.1

a.

I. Hash Code

Using same weighted algorithm, we can find the hash code for the word ‘code’:

|  |
| --- |
| =1\*99+ 2\*111+ 3\*100+ 4\*101  = 99+ 222+300+404  =1025 |

II. (I) in modulus 17:

|  |
| --- |
| 1025 mod 17 = 8 |

III. Hash Algorithm and Modulus Operator:

Hash algorithm often involves the modulus operator because it helps in limiting the output range of hash function to a specific range or size. This is important because of following reasons:

Range Restriction: Use modulo on arrays or hash tables to keep hash codes inside a range.

Distribution: A hash table can allocate space more evenly and reduce collision likelihood by limiting its hash code range.

Efficiency: Use computationally efficient modulus techniques to reduce results from many inputs.

Consistency: A constant modulus in modulo ensures consistency and that the hash table size won't change between implementations and platforms.

b. Hash Codes and Integrity of Digital Evidence

Algorithms generate accurate numerical strings called hash values for files. Recalculating the hash after file changes produces a different value. Hash code allows to access the evidence to determine if it adheres to Daubert Standards. A hash code allows to determine if data has been altered. It provide quick and easy way to analyse files. A hash code has following characteristics:

Deterministic: The specific input will always deliver the same hash values as output.

The odds of collision are low: It's unlikely that two files will have the same hash result.

Easy to calculate: A hash code can be calculated very quickly.

Change in Input changes Output: Any change in input will change the output hash value.

Question No. 2

1. Algorithm for the Provided Problem
2. Initialization

* Create a Proverb\_dictionary with proverbial phrases missing the last word as keys and their corresponding missing words values.

1. Show Flashcard Function

Modify the show\_flashcard() function

* Randomly select a proverbial phrase from proverb\_dictionary.
* Display the incomplete phrases to the user.
* Prompt the user to enter missing values
* Check if user’s input matches to the correct missing words.
* Provide feedback to the user.

1. Interactive loop

Implement an interactive loop where user can:

* Request a flashcard.
* Quit the Program.

1. Testing Approach

These steps would be followed to test the program:

Request Flashcards:

Input s will be give multiple times to request flashcards and check if the program displays incomplete phrases correctly.

Provide Correct Answers:

For each flashcard shown, correct missing word will be entered and verify if the program congratulates user.

Provide Incorrect Answers:

Incorrect missing words will be entered to check if program correctly identifies and displays the correct answer.

Quit Option:

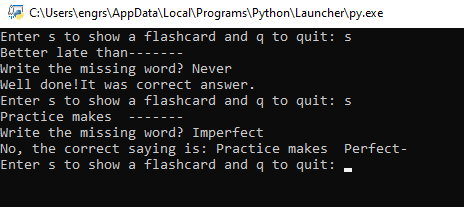
Finally, input q will quit the program and ensure exit.

1. Implementation

Modified Python code:

|  |
| --- |
| from random import \*  def show\_flashcard():    random\_proverb = choice(list(proverbs\_dictionary))  missing\_word = proverbs\_dictionary[random\_proverb]  incomplete\_phrase = random\_proverb.replace('------', '------')  print(incomplete\_phrase)  user\_input = input("Write the missing word? ")  if user\_input.lower() == missing\_word.lower():  print("Well done!It was correct answer.")  else:  print(f"No, the correct saying is: {random\_proverb.replace('------', missing\_word)}")  # Setting up the proverbs\_dictionary  proverbs\_dictionary = {  "Better late than-------": "Never",  "Too many cooks spoil the ------": "broth",  "Practice makes -------": "Perfect",  "The early bird catches the ------": "worm",  "Rome wasn't built in a ------": "day",  "Honesty is the best -------": "Policy",  }  # The interactive loop for the program  exit\_flag = False  while not exit\_flag:  user\_input = input('Enter s to show a flashcard and q to quit: ')  if user\_input.lower() == 'q':  exit\_flag = True  elif user\_input.lower() == 's':  show\_flashcard()  else:  print('You need to enter either "s" or "q".') |

Output:



1. Reflection

How: The problem description was read carefully and divided into manageable parts. I referred to the Python file I was given and the module algorithm examples to understand the program's structure. After that, I focused on implementing each algorithm component and analysing each function to ensure it worked as expected.

Recourses I studies block 3, which contained algorithm implementations and a Python introduction. I consulted the Python handbook for clarification on concepts and functions. It explained syntax and procedures.

Challenges: I had trouble making sure the program accepted user input properly, especially when prompted for submissions or responses.

Lesson Learned: I felt more confident in designing and developing interactive Python apps after this redesigned flashcard challenge. My experience taught me the necessity of iteratively testing and debugging code and breaking down challenging jobs. Practice helped me handle user input and program flow under diverse scenarios. After learning interactive programming, I feel more prepared for future professions that require it.

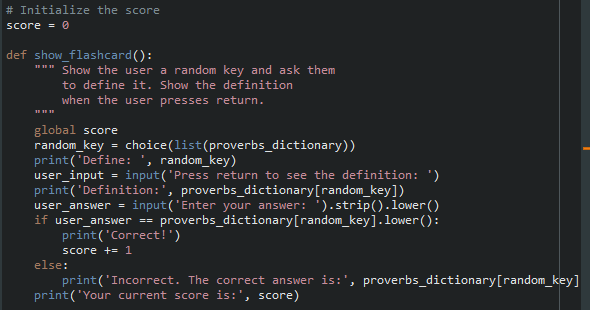
Question No. 2  
  
Part 2 (b)

The small extension would be introducing the score system to the existing flashcard program. This could track the number of correct answers by the users.

Sub Problems:

1. To implement this extension, sub problems would be:
2. Creating a variable to track the user’s score of correct answers.
3. Updating the score each time the user provide a correct answer.
4. Displaying the user’s score after each flashcard interaction or at the end of each session.
5. Adding an option to implement a feature to reset the score for each new session.

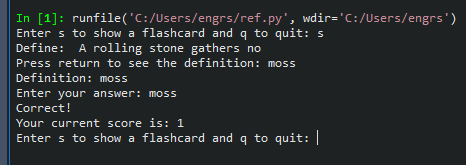
From random import choice



# set up the proverb\_dictionary

# the iterative loop

Output:



The overall effect of this modification is to incorporate a gamification component into the software, which will encourage users to participate more actively and ultimately lead to improvements in their performance.

Question No. 3

1. Carol and Dan can use an asymmetric encryption system to exchange messages safely and securely without any requirement of sharing a symmetric encryption. Here is how they can do this:
2. Carol will generate a random session key that is symmetric encryption key and encrypts the e-book file using this key.
3. Then carol can encrypt the session key using Dan’s Public key that is asymmetric encryption. Only Dan can decrypt this message with his corresponding private key.
4. Carol sends both the encrypted e-book file and the encrypted session key to Dan.
5. Dan decrypts the session key using his private key.
6. Dan then uses this decrypted session key to decrypt the e-book file.

However, due to the session key's uniqueness and the fact that it should be produced randomly for security reasons, they cannot reuse it for subsequent exchanges. Session key cannot be reused.  
Asymmetric encryption increases security by preventing anyone from decrypting the session key without Dan's private key. Because it prevents important interceptions during transmission.   
  
  
  
b.

1. 16 bit Encryption System

We have the formula, where

n = number of bits.

A 16-bit encryption scheme has = 65,536 possible keys.

1. 22 bit Encryption System

Using same formula;

A 22-bit encryption scheme has = 4,194,304 possible keys.

1. Secure Encryption System

To find out how much more secure than 22-bit system is compared to the 16-bit system, we can calculate the ratio of the number of possible keys in the 22-bit system to the number of possible keys in the 16-bit system.

Calculating ratio:

Hence, 22-bit system is 64 times more secure than 16-bit system.

c

1. User Logging

Hashing would have an appropriate role for securely storing of passwords on the server of online shopping website. When User login the website, their entered passwords can be hashed and compared with the stored hash. This ensures that the actual password remains secure even if the server is compromised.

1. Checking for Data Corruption

During the file transmission over a network, encryption would be appropriate for data corruption checking. Encryption ensures data integrity and ensures that it cannot be altered in transit. Hashing is not beneficial alone because it is a one way process and cannot ensure data integrity.

1. Preventing Piracy

The piracy of electronic console games can be prevented by encryption. The game files are encrypted so that only approved users with the decryption key can access and play the games. Since hashing is permanent and cannot protect the privacy of the game data, it is not a good choice for this situation.   
  
  
Question No. 4

Introduction

End-to-end encryption is a method of secure communication that prevents third parties to access data while it is transferred from one system or device to another.

End-to-end encrypted data can only be deciphered by the allocated receiver after being encrypted on the sender's device. No one, whether an ISP, application service provider, hacker, or other organization, may intercept or change the message while it is being transported.

Findings of the Study: The BSR study highlights how crucial it is to weigh security and privacy considerations with reference to E2EE (Lily Hay).

Recommendation: IT companies and civic society should collaborate, be transparent, and educate users to prevent E2EE exploitation.

Meta’s Action: The response of Meta to these recommendations is important in addressing concerns surrounding E2EE implementation.

Law Enforcement: The government and US law enforcement shows apprehension regarding E2ee hindering criminal investigation and advocate for backdoor access.

Perspective to Support Report Findings: Those who support end-to-end encryption argue that it safeguards human rights, privacy, and the right to free speech, while those who oppose it argue that it puts public safety at risk.   
Conclusion: The E2EE dispute has brought to light the necessity of a comprehensive strategy that assesses the advantages of the technology in terms of safeguarding individuals' privacy against the challenges posed by law enforcement. An investigation into the effects that end-to-end encryption on social media platforms will have on law enforcement, privacy, and security is of the utmost importance.

Question No. 5

a

1. The Spokesperson for the Cyber Up campaign is surprised because they expected an increase due to surge in cybercrime and fraud during pandemic.
2. The Government usually handles offences by splitting them into cyber-dependent crimes. That target digital devices and networks and other crimes where digital devices are not pivotal to the offense such as fraud.
3. As evidence of the limitations of the Computer Misuse Act, the spokesman for CyberUp draws the conclusion that the Act is flawed. He cites the gap between the rise in cybercrime and the reduction in prosecutions as evidence of the Act's shortcomings.

b.

i. High-tier Open Systems Interconnection (OSI) applications are targeted by attacks that leverage software programme or service flaws. These attacks exploit flaws in software and protocols that allow apps to connect over networks.   
  
ii. The majority of attacks against the application layer are characterised as "cyberattacks" or "network attacks," and their primary objective is to exploit vulnerabilities in the application layer of the OSI model.   
  
iii.The Hypertext Transfer Protocol (HTTP) is the most common target of these attacks, especially in web-based applications and services.   
  
iv. WAFs, or web application firewalls, mitigate application layer threats. Web Application Firewalls (WAFs) can screen out dangerous requests and prevent SQL injection and XSS. Additionally, it can analyse online application traffic arriving and leaving.

c.

Most likely, the former worker was charged under Section 3 of the 1990 Computer Misuse Act (CMA). This provision protects against computer-interfering operations.   
  
An IT staffer's aggressive application layer attack impacted the college's payment system, staff database, and intranet for a long time. The person intentionally disrupted and assaulted the organization's computer systems, violating Section 3 of the Communication Management Act.   
  
The comprehensive computer forensics study, which examined network traffic, mobile device data, and surveillance camera footage, certainly supported Section 3 of the CMA claim. This provision is best for addressing the former employee's charges because it directly tackles illegal action with the purpose to disrupt computer systems.