DEPARTMENT OF COMPUTING TECHNOLOGIES
SCHOOL OF COMPUTING
College of Engineering and Technology
SRM Institute of Science and Technology

MINI PROJECT REPORT

ODD Semester, 2023-2024

Lab code & Sub Name : 21CSS201T & Computer Organization and Architecture

Year & Semester : II & III

Project Title : SIMULATE A WORD DIVIDER

Lab Supervisor **: Mr.S.SAMINATHAN**

Team Members : 1. S GOKUL (Reg No: RA2211003011996)
2.MUTHUMANI J D (Reg No: RA2211003012002)
3.R LOKESHWARAN (Reg No: RA2211003012018)

Particulars	**Max. Marks**	**Marks Obtained** **Name:** **Register No :**
Program and Execution	20	
Demo verification &viva	15	
Project Report	05	
Total	**40**	

Date :

Staff Name :

Signature :

SIMULATE A WORD DIVIDER

OBJECTIVE:

The primary objective of simulating a word divider is to create a computational tool or algorithm that can effectively process and analyze textual content by breaking it down into discrete linguistic units, typically words. This simulation aims to accurately identify and segment words based on defined criteria, which may include spaces, punctuation, or language-specific rules. An important goal is to make this word divider language-independent, capable of handling multiple languages and character encodings, ensuring its versatility and applicability in diverse linguistic contexts. Furthermore, the simulation involves exploring various word divider algorithms, including rule-based approaches, statistical methods, and machine learning techniques, with the aim of developing efficient and accurate algorithms. The practical applications of this simulated word divider are diverse, including text indexing, search engine optimization, document summarization, and machine translation, where reliable word division within textual content is crucial. It also seeks to address challenges associated with word division, such as handling languages without clear word boundaries and adapting to specialized domains with specific linguistic requirements. Moreover, this project aims to drive innovation in the field of text processing and natural language processing by continuously improving the accuracy, efficiency, and adaptability of word divider algorithms. It sets the stage for future research in word division, encouraging the exploration of advanced algorithms and their integration with emerging technologies like deep learning, with a focus on enhancing language analysis and text processing capabilities.

ABSTRACT:

Word dividers play a vital role in natural language processing, text analysis, and information retrieval systems. They are essential tools for breaking down textual content into manageable linguistic units. This abstract explores the concept and simulation of word dividers, highlighting their significance in various domains. It begins by introducing the need for efficient word dividers to handle text-based data and emphasizes their relevance in applications like search engines, sentiment analysis, and machine learning. The abstract then delves into the components and functionalities of word dividers, discussing how they identify and separate words based on spaces, punctuation, or other defined criteria. Additionally, it covers the diverse algorithms used for word division, such as rule-based approaches, statistical methods, and machine learning techniques, while highlighting the advantages and trade-offs of each. The abstract also outlines the practical applications of word dividers in text indexing, search engine optimization, document summarization, and machine translation, demonstrating their pivotal role in enabling text analytics and natural language processing tasks. Lastly, it touches on the challenges faced in word dividing, such as handling languages without explicit word boundaries, and suggests future directions for research, including the development of advanced algorithms and their integration with emerging technologies like deep learning. In conclusion, word dividers are critical tools for language processing and text analysis, demanding ongoing research and innovation to

enhance the accuracy and efficiency of word division algorithms across a wide range of textual content.

INTRODUCTION:

The introduction to a word divider simulation project underscores the pivotal role of this computational tool in text processing and language analysis, with a focus on its applications. It can be summarized as follows:

In the landscape of text processing, natural language analysis, and information retrieval systems, the word divider stands as a foundational and indispensable component. Its primary mission is to deconstruct textual content into discrete linguistic units, commonly referred to as words, facilitating the efficient processing and analysis of textual data. This simulation project is dedicated to the exploration and development of a word divider algorithm capable of precisely identifying and segmenting words within text. An essential aspect of this endeavor is the creation of a language-independent tool that can adapt to various languages and character encodings, ensuring versatility across diverse linguistic contexts.

The practical applications of such a versatile word divider are far-reaching. It plays a pivotal role in the domain of text processing, enabling effective search engine optimization, text indexing, document summarization, and machine translation. By providing reliable word division within textual content, it supports key applications like sentiment analysis, topic modeling, and document categorization. Moreover, the project addresses the inherent challenges of word division, including the complexities of languages with ambiguous word boundaries or specialized domain-specific linguistic requirements.

This simulation project also has its sights set on fostering innovation and continual refinement in the field of word divider algorithms. It paves the way for ongoing research, encouraging the integration of advanced techniques such as deep learning to enhance language analysis and text processing capabilities. In essence, this undertaking recognizes the pivotal role of word dividers as the cornerstones of text-based data analysis, anticipating their ever-expanding applications and contributions to the evolving landscape of natural language processing and information retrieval.

HARDWARE/SOFTWARE REQUIREMENTS:

The successful implementation of a word divider simulation project relies on the harmonious integration of both hardware and software components. Hardware requirements encompass the foundational infrastructure needed to support the computational tasks and algorithm execution. This typically includes a standard computer system with a capable processor, sufficient RAM, and ample storage space to handle the simulation's computational load effectively. A modern computer system, whether desktop or laptop, is typically suitable for this purpose.

On the software front, the project necessitates a development environment that supports programming and algorithm design. Integrated Development Environments (IDEs) like Python-based Jupyter Notebook or dedicated platforms such as Visual Studio Code are commonly employed for algorithm development. Additionally, programming languages like Python, Java, or C++ are essential tools for implementing and testing the word divider algorithms. These languages provide a foundation for algorithm coding, data manipulation, and simulation execution.

The choice of software libraries and frameworks also plays a critical role in this project, as they facilitate various aspects of text processing and linguistic analysis. For example, Natural Language Processing (NLP) libraries such as NLTK (Natural Language Toolkit) in Python or Stanford NLP can be indispensable for linguistic tasks. Furthermore, the use of version control systems like Git and platforms like GitHub for collaborative development and project management enhances the efficiency and organization of the simulation project.

In summary, the successful execution of the word divider simulation project necessitates a standard computer system with adequate hardware specifications and a suitable programming environment, including IDEs, programming languages, and essential software libraries and tools. This combination of hardware and software resources forms the foundation for the development, testing, and refinement of word divider algorithms, ultimately contributing to the advancement of text processing and natural language analysis capabilities.

CONCEPTS/WORKING PRINCIPLE

The concept and working principle of a word divider are best illustrated through a block diagram, which serves as a visual representation of how this computational tool functions. The block diagram provides a clear overview of the key components and their interconnections.

At its core, a word divider operates by taking an input text, typically in the form of a string, and processing it through a series of interconnected components. The first block in the diagram represents the "Input Text," which serves as the starting point. This text is then fed into the "Word Divider Algorithm" block. The word divider algorithm is the heart of the system, responsible for identifying and segmenting words within the input text. It employs a set of rules, criteria, or machine learning techniques to achieve this task.

Once the algorithm has processed the input text, the segmented words are directed to the "Output Words" block, which represents the resulting linguistic units. The "Output Words" can be further utilized for various applications, including text indexing, search engine optimization, document summarization, and machine translation, as previously mentioned.

Importantly, the "Language and Encoding Support" block is an integral part of the system, ensuring that the word divider can handle multiple languages and character encodings. This feature enhances the versatility of the tool, making it adaptable to diverse linguistic contexts.

The overarching working principle of a word divider involves the efficient application of the word divider algorithm to the input text, resulting in the accurate segmentation of words. This segmentation process, as depicted in the block diagram, forms the basis for a wide range of text processing and language analysis applications. Overall, the block diagram provides a

comprehensive representation of the concept and working principle of a word divider, showcasing how it transforms raw textual content into structured linguistic units ready for further analysis and application.

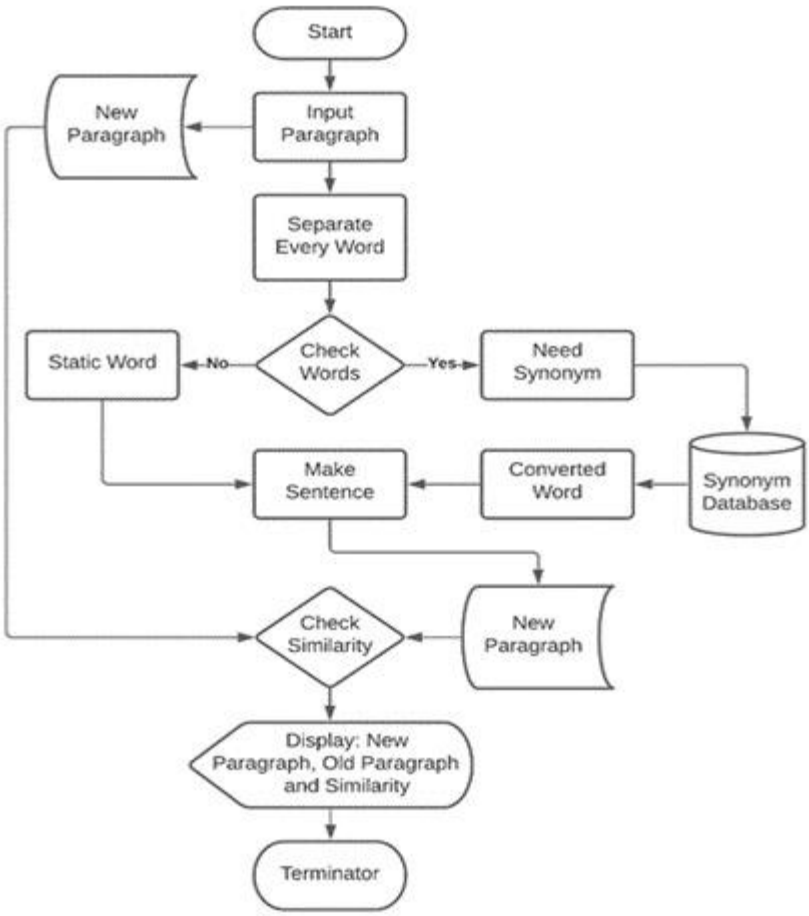
APPROACH/METHODOLOGY/PROGRAMS:

CODE :

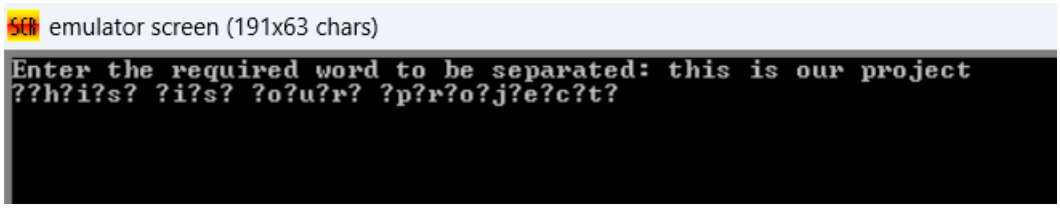
```
ORG 100h  ; Set the origin to 100h (default for DOS COM files)

MOV AH, 09h  ; Function 09h - Display String
MOV DX, offset hello_msg  ; Load address of hello_msg into DX register
INT 21h  ; Call DOS interrupt to display the prompt message

MOV AH, 0Ah  ; Function 0Ah - Read String
MOV DX, offset input_buffer  ; Load address of input_buffer into DX register
INT 21h  ; Call DOS interrupt to read a string

; Start a new line before displaying characters
MOV AH, 02h  ; Function 02h - Display Character
MOV DL, 0Dh  ; Carriage return
INT 21h  ; Call DOS interrupt to display the carriage return

MOV DL, 0Ah  ; Newline character
INT 21h  ; Call DOS interrupt to display the newline

; Display individual characters from input_buffer
MOV SI, offset input_buffer + 2  ; Load address of input_buffer into SI register (skip the length byte)

next_char:
MOV AL, [SI]  ; Move character from memory location pointed by SI into AL register

; Check if AL is the null terminator (end of string)
CMP AL, 0
JE terminate

; Display character
MOV AH, 02h  ; Function 02h - Display Character
MOV DL, AL  ; Load the character to DL register
INT 21h  ; Call DOS interrupt to display the character

; Display a question mark as the separator
MOV DL, '?'  ; Load the question mark character to DL register
INT 21h  ; Call DOS interrupt to display the question mark

INC SI  ; Increment SI to point the next character in input_buffer

JMP next_char  ; Jump to process the next character
```

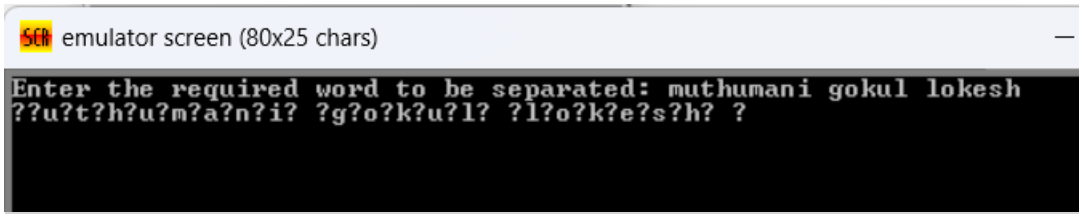
```
; After displaying all characters and question marks, start a new line
terminate:
MOV AH, 02h  ; Function 02h - Display Character
MOV DL, 0Dh  ; Carriage return
INT 21h  ; Call DOS interrupt to display the carriage return

MOV DL, 0Ah  ; Newline character
INT 21h  ; Call DOS interrupt to display the newline

MOV AH, 4Ch  ; Function 4Ch - Terminate Program
INT 21h  ; Call DOS interrupt to terminate the program

hello_msg DB 'Enter the required word to be separated: $'

; Define a buffer to store the input word (adjust size as needed)
input_buffer DB 80, ?, 80 DUP (0)  ; Declare a buffer with room for input and a length
byte
```

FLOWCHART:



OUTPUT:





CONCLUSIONS:

In conclusion, the simulation of a word divider underscores the critical importance of this computational tool in the field of text processing and natural language analysis. Through the exploration of word divider algorithms and the development of language-independent systems, this project has illustrated the versatility and adaptability of word dividers in handling diverse linguistic contexts and character encodings. The practical applications of word dividers, including text indexing, search engine optimization, document summarization, and machine translation, have been highlighted as pivotal in modern text-based data analysis. Moreover, by addressing the challenges associated with word division and promoting innovation in algorithm development, this project lays the foundation for ongoing research in the domain. The integration of advanced techniques, such as deep learning, further enhances the capabilities of word dividers, contributing to the evolving landscape of natural language processing and information retrieval. Ultimately, the word divider simulation project underscores the enduring significance of this tool in transforming raw textual content into structured linguistic units, with far-reaching implications for language analysis, text processing, and linguistic research.

REFERENCES:

https://sq.wikipedia.org/wiki/Emu8086

https://github.com/topics/emu8086