**COS10007 – Developing Technical Software  
Week 4 – Lab 4   
Student ID : 103234103 – Son Nguyen  
  
//Q1 - Explain the below//**

**//Data Structure - specialised format for organising , processing , retrieving and storing data**

**Differences bewteen linear and non linear structure - Their order is the main difference bewtween them , linear data structure are ordered whilest non linear not structured. Non linear data structure is also a more efficient way to order the data and save memory more efficiently.**

**Stacks with diagrammatic representation - A stack is a data structure that allow items to be added or removed from one end. Diagrammatic presentation is the statistical data in the form of a diagram and for push and pop meaning placing something on top of the stack and then taking the top thing off the stack address - https://www.tutorialspoint.com/data\_structures\_algorithms/images/stack\_representation.jpg.**

**Difference between linked list and stack - A stack follows the LIFO principle (last in and first out) which mean that insertion or deletion of input data can only be taken from one end - while a link list; insertion or deletion can take place from any position**

**Application of stack - To use to backtrack or to use as a memory management.**

//Question 2 - Writing a program that input lines of text and use a stack to print//

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct stack {

char letter;

struct stack \*next;

} Stack;

typedef Stack \*stackPtr;

stackPtr push(stackPtr topPtr, char c) {

stackPtr newPtr = malloc(sizeof(Stack));

if (newPtr == NULL) {

fprintf(stderr, "Memory allocation error\n");

exit(EXIT\_FAILURE);

}

newPtr->letter = c;

newPtr->next = topPtr;

return newPtr;

}

stackPtr pop(stackPtr topPtr, char \*c) {

if (topPtr == NULL) {

fprintf(stderr, "Stack underflow error\n");

exit(EXIT\_FAILURE);

}

\*c = topPtr->letter;

stackPtr tempPtr = topPtr;

topPtr = topPtr->next;

free(tempPtr);

return topPtr;

}

void reverseString(char \*str) {

stackPtr topPtr = NULL;

int len = strlen(str);

for (int i = 0; i < len; i++) {

topPtr = push(topPtr, str[i]);

}

for (int i = 0; i < len; i++) {

topPtr = pop(topPtr, &str[i]);

}

}

int main() {

char str1[20];

printf("Enter your text: ");

fgets(str1, 20, stdin);

str1[strcspn(str1, "\n")] = '\0'; // remove trailing newline character

reverseString(str1);

printf("The string in reverse: %s\n", str1);

return 0;

}  
  
Example Output   
A screenshot of a computer

Description automatically generated with medium confidence

3/  
 if (isPalindrome(str1)) {

printf("The string is a palindrome.\n");

} else {

printf("The string is not a palindrome.\n");

}

return 0;

}  
output code:

Text

Description automatically generated