COMPREHENSIVE ASSIGNMENT

DATA STRUCTURES AND ALGORITHMS

(2CS301)

TOPIC: Design a data structure that acts as find and replace, redo and undo functionality available with word processors. Demonstrate the same using appropriate data structure and contents from a file.

SUBMITTED BY:

NAME:AAYUSH KANSARA

ROLL NO: 19BCE002

DEPARTMENT:COMPUTER SCIENCE AND ENGINEERING

DATE OF SUBMISSION:24-11-2020

PROGRAMME FILES:

1. MAIN.c :: This File Contains All the Executable Code
2. READ.txt:: This is a text File For Taking Input
3. FINAL.txt:: Output Is Generated In This File

CONCEPT :

As the Input is to be taken from a file . So the Input File Is Scanned By a file pointer Word by word and User Is Provided The Option To Replace any particular Instance Of a word By Any other word. Then The User is provided with various options like To Write The Current Word In The File , Redo The Last Action Performed , Undo The Last Action Performed and read the Next Word From The File.

The Contents Of a file can either be stored word by word in 2D Matrix or Linked Lists. Storing The Words In a Linked Lists is a very efficient Compared to 2D Matrix as dynamic allocation in linked lists so no extra memory is wasted and traversal is also much efficient. So in this code each word is stored in the linked lists and each word represents a node in a linked lists. And the other data structure used here is Stack

LOGIC FOR USING 2 STACK:

For Performing Undo And Redo Operations The Most Efficient Data Structure Is Stack which performs undo and redo in O(1) time Complexity. In This Programme 2 Stack Is Used One Is Redo Stack And Other Is Undo Stack.

- For Writing Current Word In a File :The Current Word Is Pushed Into Undo Stack.
- For Undo Operation: Top Element/word Of Undo Stack is popped out and the popped out word is again pushed into Second Stack(Redo Stack).
- For Redo Operation: Pop The Top Element Of Stack 2 and push The Popped Out Element into Undo Stack.

TIME COMPLEXITY:

- FOR READING ALL THE WORDS FROM A FILE::>> O(N) ,where n is the number of words in the file
- For UNDO OPERATION::>> O(1),Constant time time Complexity
- For REDO OPERATION::>> O(1),Constant time time Complexity
- For REPLACING WORDS IN A FILE::>> O(N),where n is the number of words in the file

CODE :

```
/*
TOPIC:Design a data structure that acts as find and replace
redo and undo functionality available with word processors .
Demonstrate the same using appropriate data structure and contents from a file.
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 256
#define MAX 256
/* Structure for Stack 1(UNDO STACK)*/
typedef struct stack_1
{
   char data_1[SIZE];
   struct stack_1 *previous_1;
}stack_1;
/* Structure for Stack 2(REDO STACK)*/
typedef struct stack_2
{
   char data_2[SIZE];
   struct stack_2 *previous_2;
}stack_2;

/*Structure for Reversing Stack 1 And Printing The Final Output*/
typedef struct reverse{
   char info[SIZE];
   struct reverse *previous;
}reverse;
/* Assigning The Top Pointer Of All The Three Stack as NULL*/
```

```
stack_1 *topStackPointer_1 = NULL;
stack_2 *topStackPointer_2 = NULL;
reverse *topStackPointer_reverse=NULL;
/*  FUNCTION DECLARATION */
void push_1(char[]);
void push_2(char[]);

char* pop_1();
char* pop_2();

void printStack_1();
void printStack_2();
void printStack_reverse();
void replace();

int main()
{
      printf("**************************WELCOME USER
**************************\n\n");
      /* To REPLACE THE WORDS IN A FILE */
      while(1)
      {
         printf("Do You Want To Replace Words In A File\n");
         printf("Enter 1 for yes\n");
         printf("Enter 2 for no\n");
         printf("Enter Your Choice:");
         int choice;
         scanf("%d",&choice);
         if(choice==2)
         {
            break;
```

```
        }
        else{
            replace();
        }
      }

   char String[SIZE];
   /* OPENING THE FILE FOR READING THE WORDS*/
   FILE *in_file = fopen("A:\\Comprehensive Assignment\\Data Structures And Algorithms\\Dsa_Final\\Read.txt", "r");
   if (in_file == NULL)
   {
       printf("Error file missing\n");
       exit(-1);
   }
    printf("\n");
    int redo_cnt=0;//MAINTAINING THE REDO CNT AND UNDO CNT TO AVOID PROGRAMME TERMINATION
    int undo_cnt=0;
    printf("Words From The Text File Will Be Displayed On The Console You Have To Select What Operation You Want To Perform\n");
    AGAIN:
    while(fscanf(in_file, "%s", String) != EOF)//this loop searches the for the current word
    {

        printf("\nCurrent  Word Is :%s\n",String);
        int choice;
        while(choice!=4)
        {
            printf("\n");
            printf("SELECT THE OPERATION YOU WANT TO PERFORM :\n");
```

```
printf("1.WRITE THE CURRENT WORD\n");
printf("2.UNDO THE LAST ACTION PERFORMED\n");
printf("3.REDO THE LAST ACTION PERFORMED\n");
printf("4.READ THE NEXT WORD FROM THE FILE\n");
printf("ENTER YOUR CHOICE:");
scanf("%d",&choice);
printf("\n");
if(choice==1)
{
   push_1(String);
   /*WRITING THE CURRENT WORD IN A FILE*/
}
else if(choice==2)
{
   /* TO UNDO THE LAST ACTION PERFORMED*/
   char temp[20];
   strcpy(temp,pop_1());
   /* POP THE TOP ELEMENT OF THE UNDO STACK AND PUSH THE
STRING RETURNED INTO REDO STACK*/
   push_2(temp);
   undo_cnt++;
}
else if(choice==3)
{    /*IF THE UNDO COUNT IS ZERO THEN WE PUSH CURERENT STRING
INTO STACK 1*/
    if(undo_cnt==0)
    {
      push_1(String);
    }
    /* POP THE TOP STRING FROM REDO STACK(STACK 2) AND PUSH
THE POPPED OUT STRING INTO STACK 1 */
```

```
                else
                {
                    char tmp[20];
                    strcpy(tmp,pop_2());
                    push_1(tmp);
                }
                redo_cnt++;

            }
            else if(choice==4)
            {
                printf("Reading The Next Word From The File\n");
                choice=0;
                /* READING THE NEXT WORD */
                 goto AGAIN;
            }

        }

    }
    printStack_1();/*Reversing The Stack 1 And Storing All The Contents In the Reverse Stack*/
    printStack_reverse();/*Printing the Reverse Stack and Print All The Contents In The Output File*/
    printf("\n**********The Programme Is Closed**********\n\n");
    printf("\n**********THANKKYOU**********\n");
}
void replace()
{
    FILE *fp1, *fp2,*fpreader,*fpreader_1;
```

```
char word[SIZE];
char string[SIZE], replace[SIZE];
char temp[] = "temp.txt", *ptr1, *ptr2;
char fname[]="Read.txt";
printf("\n");
printf("Printing The Contents Of A File:\n");
 fpreader = fopen(fname, "r");
char f;
f = fgetc(fpreader);
while (f != EOF)
{
   printf ("%c", f);
   f = fgetc(fpreader);
}

fclose(fpreader);
 printf("\n");
/* get the word to delete from the user */
printf("Enter the word to be replaced:");
scanf("%s", word);

/* get the word to replace */
printf("Enter your replace word:");
scanf("%s", replace);

/* open input file in read mode */
fp1 = fopen(fname, "r");

/* error handling */
if (!fp1) {
```

```
        printf("Unable to open the input file!!\n");
        return;
}

/* open temporary file in write mode */
fp2 = fopen(temp, "w");
/* error handling */
if (!fp2) {
        printf("Unable to open temporary file!!\n");
        return;
}

/* delete the given word from the file */
while (!feof(fp1)) {
        strcpy(string, "\0");
        /* read line by line from the input file */
        fgets(string, MAX, fp1);

        /*
         * check whether the word to delete is
         * present in the current scanned line
         */
        if (strstr(string, word)) {
                ptr2 = string;
                while (ptr1 = strstr(ptr2, word)) {
                        /*
                         * letters present before
                         * before the word to be replaced
                         */
                        while (ptr2 != ptr1) {
```

```
                fputc(*ptr2, fp2);
                ptr2++;
            }
            /* skip the word to be replaced */
            ptr1 = ptr1 + strlen(word);
            fprintf(fp2, "%s", replace);
            ptr2 = ptr1;
        }

        /* characters present after the word to be replaced */
        while (*ptr2 != '\0') {
            fputc(*ptr2, fp2);
            ptr2++;
        }
    } else {
        /*
         * current scanned line doesn't
         * have the word that need to be replaced
         */
        fputs(string, fp2);
    }
}

/* close the opened files */
fclose(fp1);
fclose(fp2);

/* remove the input file */
remove(fname);
/* rename temporary file name to input file name */
```

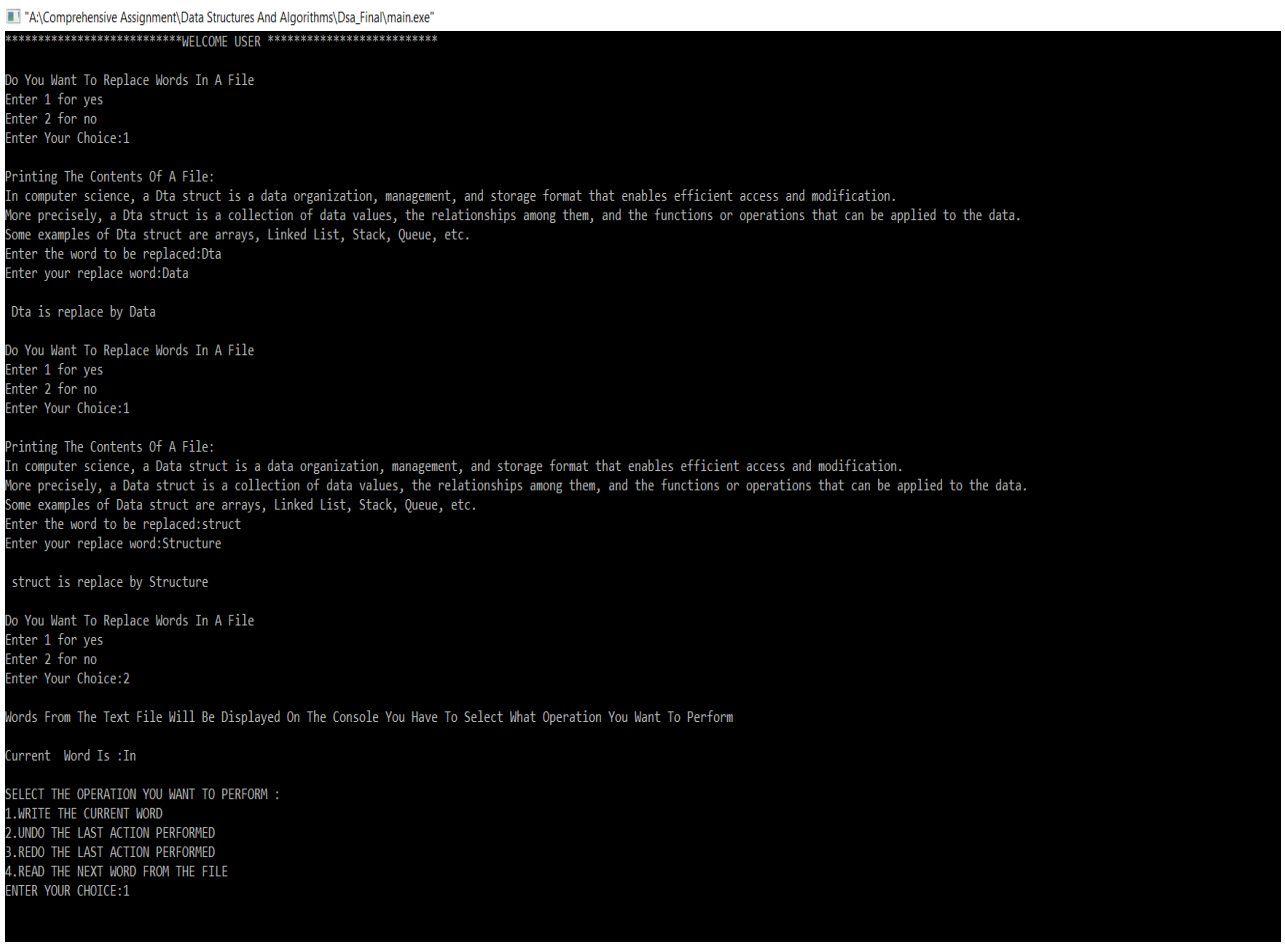
```
        rename(temp, fname);
        printf("\n");
        return ;
}
/*PUSH THE WORD INTO THE STACK 1 */
void push_1(char data[])
{
            stack_1 *NewNodePointer = malloc(sizeof(stack_1));
            strcpy(NewNodePointer->data_1,data);
            NewNodePointer->previous_1 = topStackPointer_1;
            topStackPointer_1 = NewNodePointer;
            return;
}
/*PUSH THE WORD INTO THE STACK 2 */
void push_2(char data[])
{
            stack_2 *NewNodePointer = malloc(sizeof(stack_2));
            strcpy(NewNodePointer->data_2,data);
            NewNodePointer->previous_2 = topStackPointer_2;
            topStackPointer_2 = NewNodePointer;
            return;
}
/* REVERSING THE STACK 1 AND COPYING ALL THE DATA INTO REVERSE
STACK*/
void printStack_1(void){
            stack_1 *TempPointer = topStackPointer_1;

            while(TempPointer != NULL)
    {
        reverse *NewNodePointer = malloc(sizeof(reverse));
        strcpy(NewNodePointer->info,TempPointer->data_1);
```

```
        NewNodePointer->previous = topStackPointer_reverse;
          topStackPointer_reverse = NewNodePointer;
           TempPointer = TempPointer->previous_1;
            };
            printf("***************\n");
            return;
}
/* PRINTING THE STACK 2*/
void printStack_2()
{
    printf("***************\n");
      stack_2 *TempPointer = topStackPointer_2;
      while(TempPointer != NULL)
    {
                    printf("%s ",TempPointer->data_2);
                    TempPointer = TempPointer->previous_2;
            };
            printf("***************\n");
            return;
}

/* Pops The Top String/Element Of Stack 1 and return the Popped String */
char* pop_1()
{
            static char str[10];
            strcpy(str,topStackPointer_1->data_1);
            stack_1 *PreviousNodePointer = topStackPointer_1->previous_1;
            free(topStackPointer_1);
            topStackPointer_1 = PreviousNodePointer;
            return str;
```

```
}
/* Pops The Top String/Element Of Stack 2 and return the Popped String */
char* pop_2()
{
        static char str[10];
        strcpy(str,topStackPointer_2->data_2);
        stack_2 *PreviousNodePointer = topStackPointer_2->previous_2;
        free(topStackPointer_2);
        topStackPointer_2 = PreviousNodePointer;
        return str;
}
/* PRINTING THE REVERSED STACK ON THE CONSOLE ND WRITING THE
OUTPUT IN THE NEW FILE*/
void printStack_reverse()
{
    reverse *TempPointer = topStackPointer_reverse;FILE *fp;
    fp = fopen ("A:\\Comprehensive Assignment\\Data Structures And
Algorithms\\Dsa_Final\\final.txt","w");
    while(TempPointer != NULL)
     {
                printf("%s ",TempPointer->info);
                fprintf (fp,"%s ",TempPointer->info);
                TempPointer = TempPointer->previous;
          };
     fclose (fp);
     printf("\n");
          printf("ALL THE CONTENTS OF A FINAL TEXT FILE ARE  WRITTEN IN A
NEW TEXT FILE\n");
        return;
}
```

----------END OF CODE----------

SCREENSHOTS:



"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :computer

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:2


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :science,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :science,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :a

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :Data

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:2


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:3


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :Structure

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:2
```

```
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :is

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :a

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :data

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :organization,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :management,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

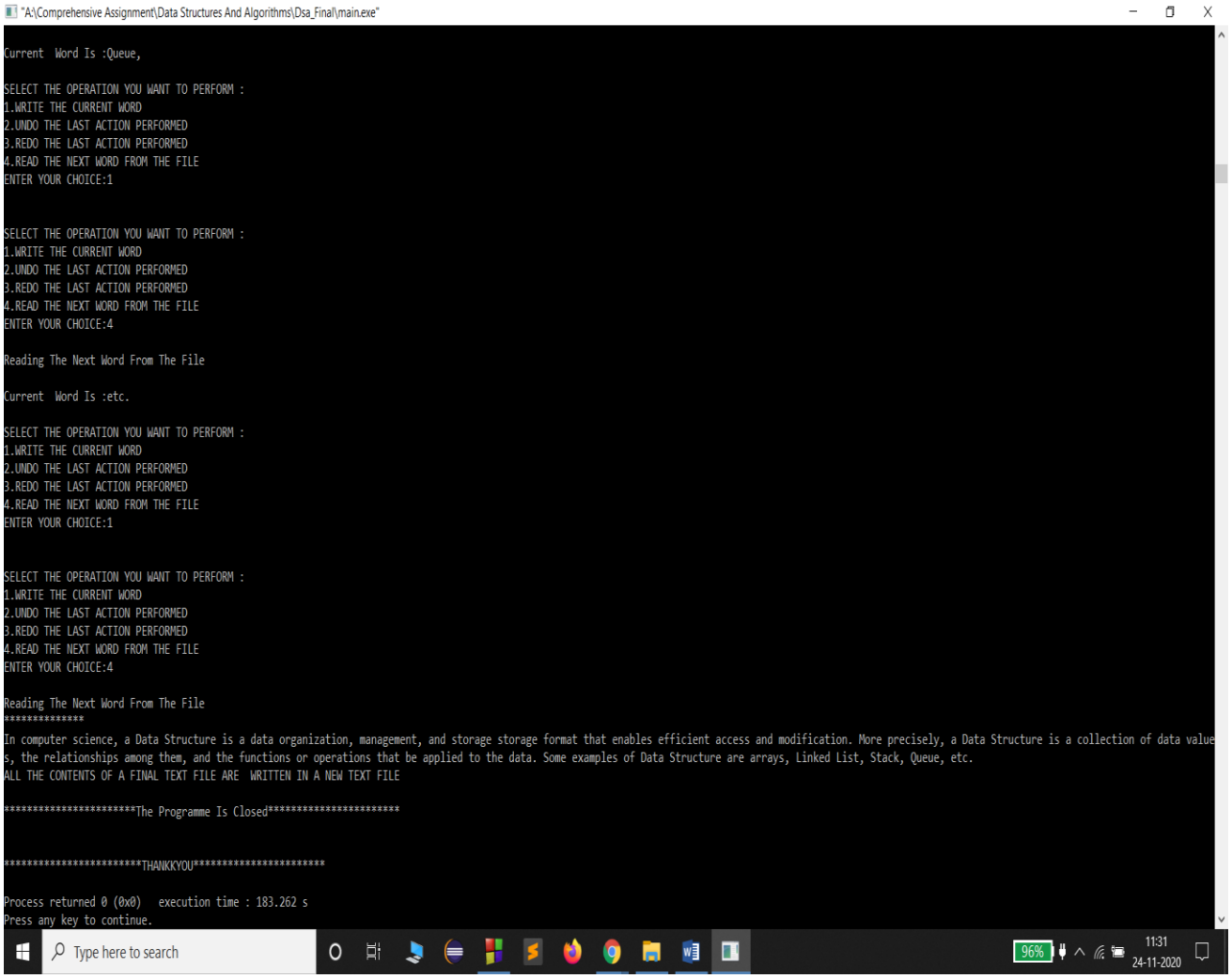
```
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :and

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :storage

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:
1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
Reading The Next Word From The File

Current  Word Is :Structure

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :are

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :arrays,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1
```

"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"

```
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4

Reading The Next Word From The File

Current  Word Is :List,

SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:2


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:3


SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4
```


"A:\Comprehensive Assignment\Data Structures And Algorithms\Dsa_Final\main.exe"
Current Word Is :Queue,
SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1
SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4
Reading The Next Word From The File
Current Word Is :etc.
SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:1
SELECT THE OPERATION YOU WANT TO PERFORM :
1.WRITE THE CURRENT WORD
2.UNDO THE LAST ACTION PERFORMED
3.REDO THE LAST ACTION PERFORMED
4.READ THE NEXT WORD FROM THE FILE
ENTER YOUR CHOICE:4
Reading The Next Word From The File
****
In computer science, a Data Structure is a data organization, management, and storage storage format that enables efficient access and modification. More precisely, a Data Structure is a collection of data value
s, the relationships among them, and the functions or operations that be applied to the data. Some examples of Data Structure are arrays, Linked List, Stack, Queue, etc.
ALL THE CONTENTS OF A FINAL TEXT FILE ARE WRITTEN IN A NEW TEXT FILE
**********The Programme Is Closed**********
**********THANKKYOU**********
Process returned 0 (0x0) execution time : 183.262 s
Press any key to continue.
Type here to search
96%
11:31
24-11-2020

----------END OF FILE ----------