Batch: P4-1 **Roll No.: 16010423076**

Experiment / assignment / tutorial No. 8

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Write a program in C to demonstrate use of a pointer.

AIM: 1) Write a program that calculates and prints the transpose of a given matrix using pointers(use function to find transpose of a matrix).
2) Write a file copy program in C that copies a file into another.

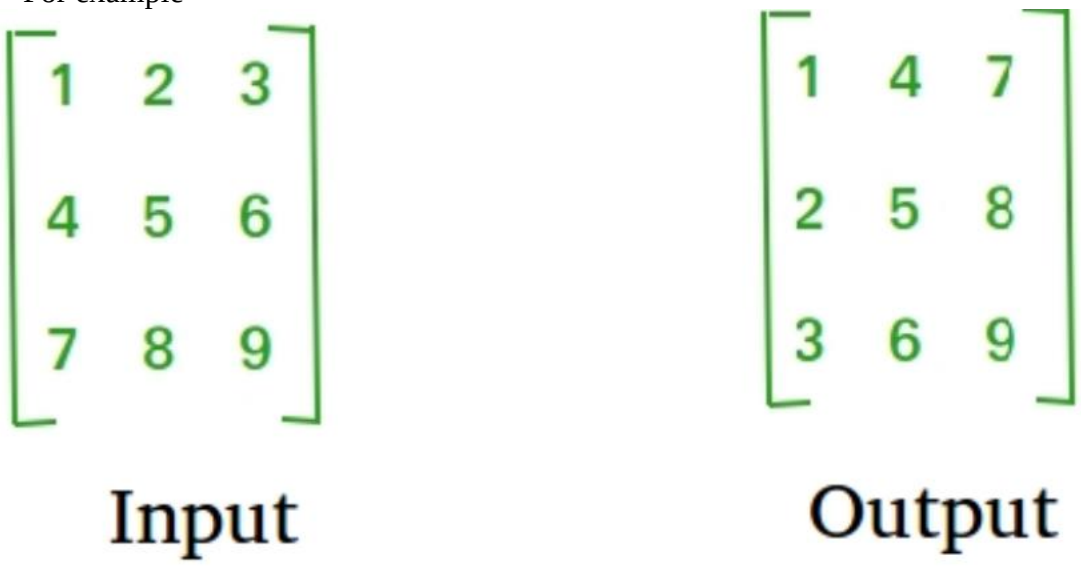
Expected OUTCOME of Experiment:
Apply concepts of pointers in dynamic memory allocation and file handling(CO5).

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.

Problem Definition:

1) The program allows the user to input a matrix, dynamically allocates memory for the matrix and its transpose, calculates and prints the transpose of the matrix using pointers, and then frees the dynamically allocated memory(Use function to find the transpose of a matrix).
   For example



2) The program copies the contents of a source file to the destination file, character by character.

Algorithm:

1)

1. Define a function transMat to calculate the transpose of a given matrix.
2. Declare variables row and col to store the number of rows and columns, respectively.
3. Prompt the user to input the number of rows and columns for the matrix.
4. Allocate memory dynamically for the matrix using a 2D array.
5. Prompt the user to input elements into the matrix.
6. Allocate memory dynamically for the transpose matrix using another 2D array.
7. Call the transMat function to calculate the transpose of the matrix.
8. Print the transpose matrix.
9. Free the dynamically allocated memory for both the matrix and its transpose to prevent memory leaks.

2)

1. Open the source file in read mode and the destination file in write mode.
2. Iterate through the source file character by character using a loop until the end of the file is reached.
3. Read each character from the source file using fgetc.
4. Write the character to the destination file using fputc.
5. Close both the source and destination files.
6. Print a message indicating that the file content has been copied successfully.
7. End the program.

Implementation details:

1)

```
#include <stdio.h>
#include <stdlib.h>

void transMat(int **mat,int **trans,int row,int col) {
   for (int i=0;i<row;i++){
      for (int j=0;j<col;j++){
         trans[j][i] = mat[i][j];
      }
   }
}

int main() {
   int row,col;
   printf("Enter rows and columns : ");
   scanf("%d %d",&row,&col);

   int **mat = (int**)malloc(row*sizeof(int *));
   for (int i=0;i<row;i++) {
      mat[i] = (int*)malloc(col*sizeof(int));
   }

   printf("Enter elements in mat form : \n");
   for (int i=0;i<row;i++) {
      for (int j=0;j<col;j++) {
         scanf("%d",&mat[i][j]);
      }
   }

   // memory allocation
   int **trans=(int**)malloc(col*sizeof(int*));
   for (int i=0;i<col;i++) {
      trans[i] = (int *)malloc(row * sizeof(int));
   }

   transMat(mat,trans,row,col);
   printf("Transpose of the matrix is :\n");
   for (int i=0;i<col;i++) {
      for (int j=0;j<row;j++) {
         printf("%d ",trans[i][j]);
      }
      printf("\n");
```

```
    }

    // freeing the memory
    for (int i=0;i<row;i++) {
        free(mat[i]);
    }
    free(mat);
    for (int i=0;i<col;i++) {
        free(trans[i]);
    }
    free(trans);

    return 0;
}
```

2)

```
#include <stdio.h>
int main() {

    FILE *srcfile,*dstfile;
    char character;

    srcfile = fopen("source.txt","r");
    dstfile = fopen("destination.txt","w");

    while((character = fgetc(srcfile)) != EOF) {
        fputc(character, dstfile);
    }

    fclose(srcfile);
    fclose(dstfile);
    printf("File content has been copied successfully...");
    return 0;
}
```

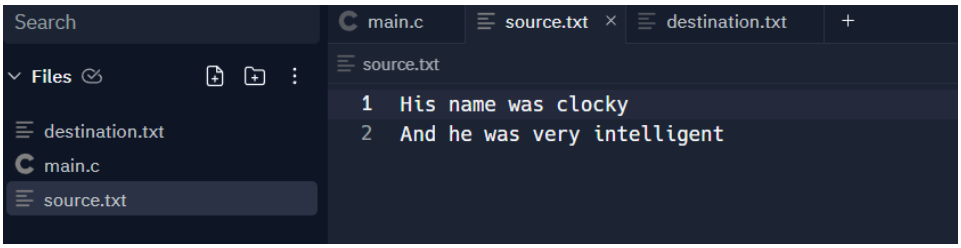
Output(s):

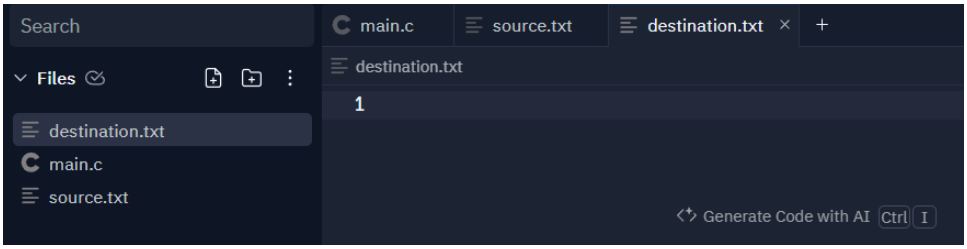
1)

```
Output

/tmp/D7MEyCqsm7.o
Enter rows and columns : 3 3
Enter elements in mat form :
1 2 3
4 5 6
7 8 9
Transpose of the matrix is :
1 4 7
2 5 8
3 6 9


=== Code Execution Successful ===
```

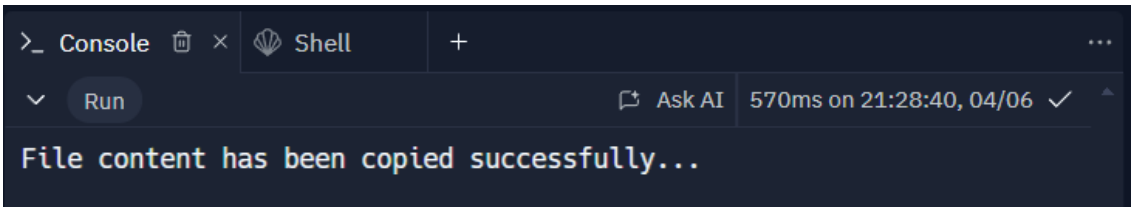
2)

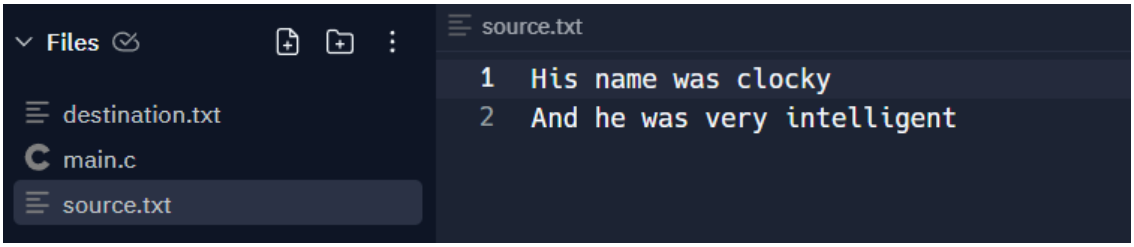
Before :

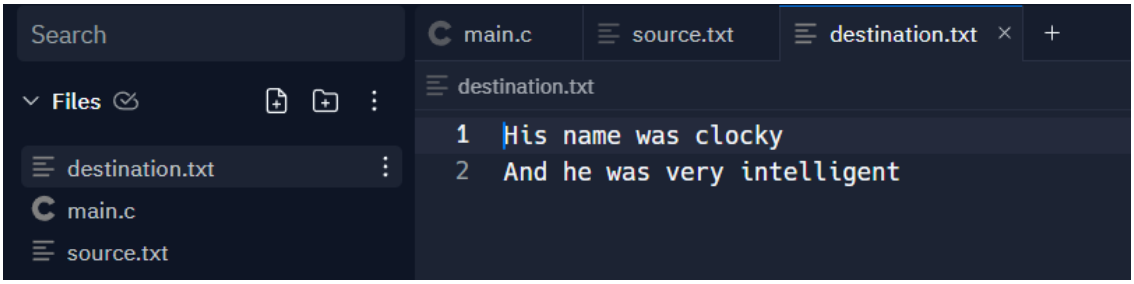




After Execution :







Conclusion:

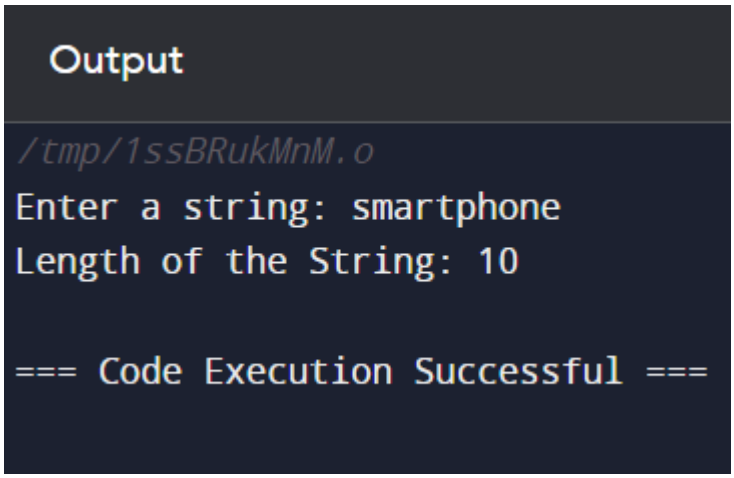
After coding these two problems, I've gained a deeper understanding of dynamic memory allocation in C and how to manipulate matrices using pointers. Additionally, I've learned how to efficiently transpose a matrix and the importance of freeing dynamically allocated memory to prevent memory leaks. Moreover, I've grasped the concept of file handling in C, especially copying contents from one file to another, which can be useful for various file manipulation tasks. Overall, these exercises have enhanced my problem-solving skills and expanded my knowledge of programming concepts in C.

Post Lab Descriptive Questions

- WAP to accept a string from the user and calculate the length of a given string using pointers.

  Input :

```
#include <stdio.h>
int len_calculator(char* inpstr) {
    int lencount= 0;
    while (*inpstr != '\0') {
        lencount++;
        inpstr++;
    }
    return lencount;
}

int main() {
    char inpstr[999];
    printf("Enter a string: ");
    scanf("%s", inpstr);
    printf("Length of the String: %d", len_calculator(inpstr));
    return 0;
}
```

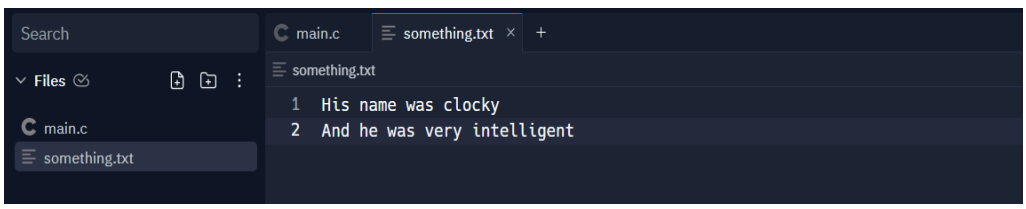


- WAP to count the number of characters and number of lines in a file.

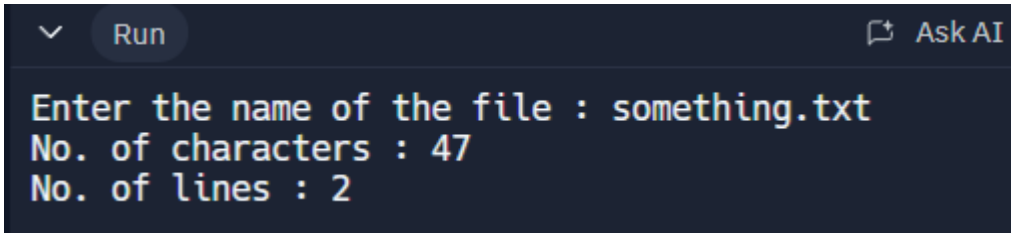
Input :

```
#include <stdio.h>
int main() {
   FILE *file;
   char filename[999];
   char ch;
   int countchar = 0;
   int countline = 0;

   printf("Enter the name of the file : ");
   scanf("%s",filename);

   file = fopen(filename, "r");
   while ((ch = fgetc(file)) != EOF) {
      countchar++;
      if (ch == '\n') {
         countline++;
      }
   }
   fclose(file);

   printf("No. of characters : %d\n",countchar);
   printf("No. of lines : %d\n",countline+1);
}
```

Text file :

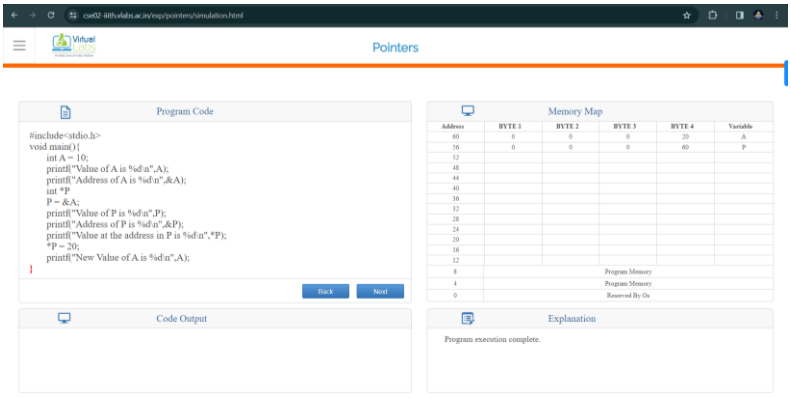


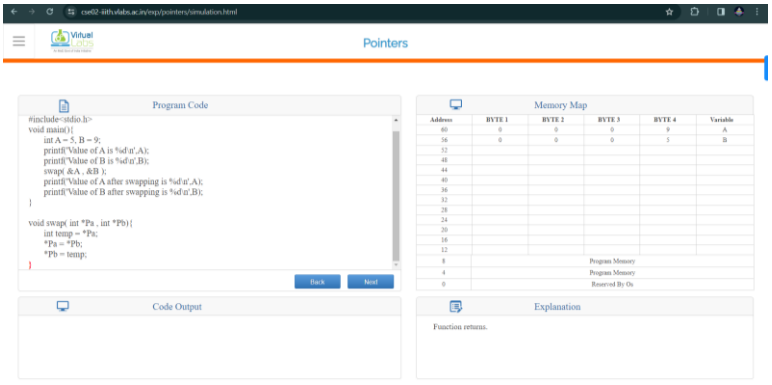
Output :



- Virtual Lab for pointers

https://cse02-iiith.vlabs.ac.in/exp/pointers/simulation.html





Date: _____ **Signature of faculty in-charge**