**Mandsaur University**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**Program Name**: B. Tech (Computer Science and Engineering)

**Assessment Name**: Assignment-5

***Semester-III***

**Name**: Aniket Purohit

**Enrollment Number**: 22ENG3CSE1009

1. Write a c program to create ,declare and initialize a detail using structure.

CODE

#include <stdio.h>

/\*structure declaration\*/

struct employee {

char name[30];

int empId;

float salary;

};

int main()

{

/\*declare and initialization of

structure variable\*/

struct employee emp = { "Mike", 1120, 76909.00f };

printf("\n Name: %s", emp.name);

printf("\n Id: %d", emp.empId);

printf("\n Salary: %f\n", emp.salary);

return 0;

}



1. C program to read and print an employee’s detail using structure.

#include <stdio.h>

struct employee

{

char name[50] ;

int empId ;

int age ;

float salary ;

} ;

int main( )

{

struct employee emp ;

printf("\n Enter employee details !\n") ;

printf("\n Name : ") ;

gets(emp.name ) ;

printf("\n ID : ") ;

scanf("%d",&emp.empId ) ;

printf("\n Age : ") ;

scanf("%d",&emp.age ) ;

printf("\n Salary : ") ;

scanf("%f",&emp.salary ) ;

printf("\n Entered employee detail are !" ) ;

printf("\n Name: %s" ,emp.name ) ;

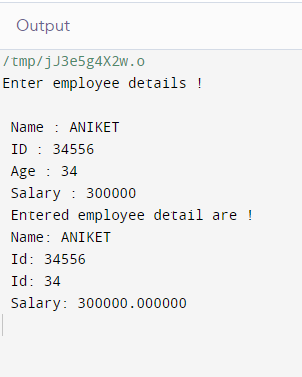
printf("\n Id: %d" ,emp.empId ) ;

printf("\n Id: %d" ,emp.age ) ;

printf("\n Salary: %f\n",emp.salary ) ;

return 0 ;

}



1. C program to demonstrate example of nested structure .

#include<stdio.h>

struct address

{

char city[20];

int pin;

char phone[14];

};

struct employee

{

char name[20];

struct address add;

};

void main ()

{

struct employee emp;

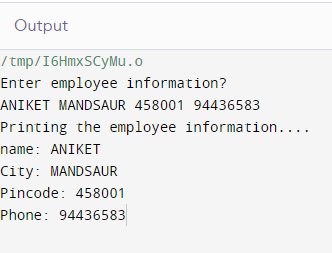
printf("Enter employee information?\n");

scanf("%s %s %d %s",&emp.name,&emp.add.city, &emp.add.pin,&emp.add.phone);

printf("Printing the employee information....\n");

printf("name: %s\nCity: %s\nPincode: %d\nPhone: %s",emp.name,emp.add.city,emp.add.pin,emp.add.phone);

}



1. C program to demonstrate example structure pointer (structure with pointer).

#include<stdio.h>

struct student{

int sno;

char sname[30];

float marks;

};

int main (){

struct student s;

struct student \*st;

printf("enter sno, sname, marks:");

scanf ("%d%s%f", & s.sno, s.sname, &s. marks);

st = &s;

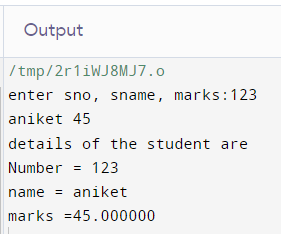
printf("details of the student are\n");

printf("Number = %d\n", st ->sno);

printf("name = %s\n", st->sname);

printf("marks =%f\n", st ->marks);

}



1. C program to demonstrate example structure pointer (structure with pointer )using user define function.

#include <stdio.h>

#include <stdlib.h>

struct Person {

char name[50];

int age;

char gender;

};

void inputPersonDetails(struct Person\* p) {

printf("Enter name: ");

scanf("%s", p->name);

printf("Enter age: ");

scanf("%d", &(p->age));

printf("Enter gender (M/F): ");

fflush(stdin);

scanf(" %c", &(p->gender));

}

void displayPersonDetails(struct Person\* p) {

printf("\nName: %s\n", p->name);

printf("Age: %d\n", p->age);

printf("Gender: %c\n", p->gender);

}

int main() {

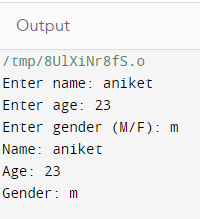
struct Person person;

inputPersonDetails(&person);

displayPersonDetails(&person);

return 0;

}



1. C program to declare, initialise an union ,example of union.

#include <stdio.h>

#include <string.h>

union Data {

int intValue;

float floatValue;

char stringValue[20];

};

int main() {

union Data data;

data.intValue = 42;

printf("Value of intValue: %d\n", data.intValue);

data.floatValue = 3.14;

printf("Value of floatValue: %f\n", data.floatValue);

strcpy(data.stringValue, "Hello, Union!");

printf("Value of stringValue: %s\n", data.stringValue);

printf("\nAfter modifying intValue:\n");

data.intValue = 100;

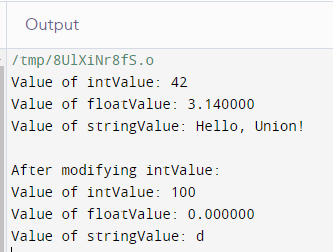
printf("Value of intValue: %d\n", data.intValue);

printf("Value of floatValue: %f\n", data.floatValue);

printf("Value of stringValue: %s\n", data.stringValue);

return 0;

}



1. C program to demonstrate example of structure of array.

#include <stdio.h>

#include <string.h>

struct Student {

char name[50];

int rollNumber;

int marks[1];

};

int main() {

struct Student students[2];

for (int i = 0; i < 2; i++) {

printf("Enter details for Student %d:\n", i + 1);

printf("Name: ");

scanf("%s", students[i].name);

printf("Roll Number: ");

scanf("%d", &students[i].rollNumber);

printf("Marks :\n");

for (int j = 0; j < 1; j++) {

printf("Subject %d: ", j + 1);

scanf("%d", &students[i].marks[j]);

}

printf("\n");

}

printf("\nStudent Details:\n");

for (int i = 0; i < 2; i++) {

printf("Student %d\n", i + 1);

printf("Name: %s\n", students[i].name);

printf("Roll Number: %d\n", students[i].rollNumber);

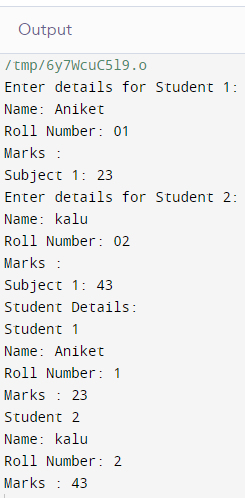
printf("Marks : %d",students[i].marks[0]);

printf("\n");

}

return 0;

}



1. C program to add two distance in feet and inches using structure.

#include <stdio.h>

struct Distance {

int feet;

float inches;

};

struct Distance addDistances(struct Distance d1, struct Distance d2) {

struct Distance result;

result.inches = d1.inches + d2.inches;

if (result.inches >= 12.0) {

result.inches -= 12.0;

result.feet = d1.feet + d2.feet + 1;

} else {

result.feet = d1.feet + d2.feet;

}

return result;

}

int main() {

struct Distance distance1;

printf("Enter the first distance:\n");

printf("Feet: ");

scanf("%d", &distance1.feet);

printf("Inches: ");

scanf("%f", &distance1.inches);

struct Distance distance2;

printf("Enter the second distance:\n");

printf("Feet: ");

scanf("%d", &distance2.feet);

printf("Inches: ");

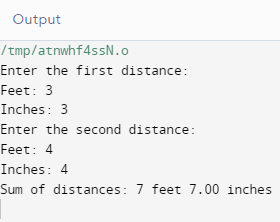
scanf("%f", &distance2.inches);

struct Distance sum = addDistances(distance1, distance2);

printf("Sum of distances: %d feet %.2f inches\n", sum.feet, sum.inches);

return 0;

}



1. C program to extract individual bytes from an unsigned int using union.

CODE

#include <stdio.h>

// Define a union to store an unsigned integer and its individual bytes

union IntBytes {

unsigned int num;

unsigned char bytes[sizeof(unsigned int)];

};

int main() {

union IntBytes intUnion;

intUnion.num = 0x12345678;

printf("Original integer: 0x%08X\n", intUnion.num);

printf("Individual bytes: ");

for (int i = 0; i < sizeof(unsigned int); i++) {

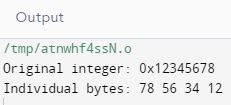
printf("%02X ", intUnion.bytes[i]);

}

printf("\n");

return 0;

}



1. Wrirte a program in c to create and stroe information in a text file.

CODE

#include <stdio.h>

#include <stdlib.h>

int main()

{

char str[1000];

FILE \*fptr;

char fname[20]="infromation.txt";

fptr=fopen(fname,"w");

if(fptr==NULL)

{

printf(" Error in opening file!");

exit(1);

}

printf(" Input a sentence for the file : ");

fgets(str, sizeof(str), stdin);

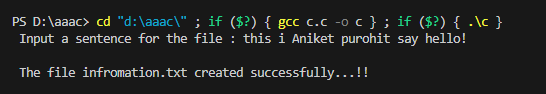
fprintf(fptr,"%s",str);

fclose(fptr);

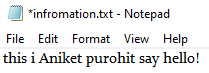
printf("\n The file %s created successfully...!!\n\n",fname);

return 0;

}



**“ information.txt”**

****

1. Write a C program to read an existing file.

#include <stdio.h>

#include <stdlib.h>

void main() {

FILE \*fptr;

charfname[20];

charstr;

printf("\n\n Read an existing file :\n");

printf("------------------------------\n");

printf(" Input the filename to be opened : ");

scanf("%s",fname);

fptr = fopen (fname, "r");

if (fptr == NULL)

{

printf(" File does not exist or cannot be opened.\n");

exit(0);

}

printf("\n The content of the file %s is :\n",fname);

str = fgetc(fptr);

while (str != EOF)

{

printf ("%c", str);

str = fgetc(fptr);

}

fclose(fptr);

printf("\n\n");

}

1. w a program in C to write multiple lines to a text file.

#include <stdio.h>

int main ()

{

  FILE \* fptr;

  inti,n;

  charstr[100];

  charfname[20]="test.txt";

  char str1;

    printf("\n\n Write multiple lines in a text file and read the file :\n");

    printf("------------------------------------------------------------\n");

    printf(" Input the number of lines to be written : ");

    scanf("%d", &n);

    printf("\n :: The lines are ::\n");

    fptr = fopen (fname,"w");

    for(i = 0; i< n+1;i++)

        {

        fgets(str, sizeofstr, stdin);

        fputs(str, fptr);

        }

  fclose (fptr);

/\*-------------- read the file -------------------------------------\*/

    fptr = fopen (fname, "r");

    printf("\n The content of the file %s is  :\n",fname);

    str1 = fgetc(fptr);

    while (str1 != EOF)

        {

            printf ("%c", str1);

            str1 = fgetc(fptr);

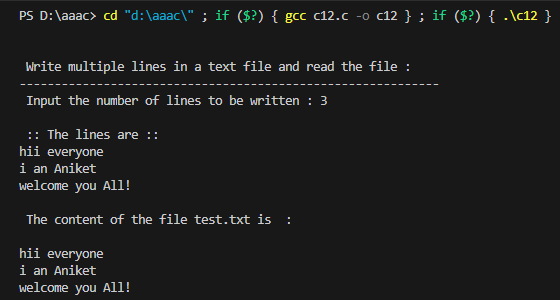
        }

    printf("\n\n");

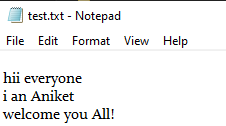
    fclose (fptr);

    return0;

}



**“test.txt”**



1. Write a program in C to read the file and store the lines in an array.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LSIZ 128

#define RSIZ 10

int main(void)

{

    char line[RSIZ][LSIZ];

    char fname[20];

    FILE \*fptr = NULL;

    int i = 0;

    int tot = 0;

    printf(" Input the filename to be opened : ");

    scanf("%s",fname);

    fptr = fopen(fname, "r");

    while(fgets(line[i], LSIZ, fptr))

    {

        line[i][strlen(line[i]) - 1] = '\0';

        i++;

    }

    tot = i;

    printf("\n The content of the file %s  are : \n",fname);

    for(i = 0; i< tot; ++i)

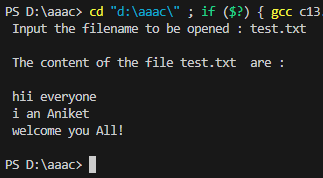
    {

        printf(" %s\n", line[i]);

    }

    printf("\n");

}



1. Write a program in C to find the content of a file and the number of lines in a text file.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define LSIZ 128

#define RSIZ 10

int main(void)

{

    char line[RSIZ][LSIZ];

    char fname[20];

    FILE \*fptr = NULL;

    Int i = 0;

    int tot = 0;

    printf("\n\n Find the content of the file and number of lines in a Text File :\n");

    printf("----------------------------------------------------------------------\n");

    printf(" Input the file name to be opened : ");

    scanf("%s",fname);

    fptr = fopen(fname, "r");

/\*--------------------- store the lines into an array ----------------\*/

    while(fgets(line[i], LSIZ, fptr))

    {

        line[i][strlen(line[i]) - 1] = '\0';

        i++;

    }

    tot = i;

    printf("\n The content of the file %s  are : \n",fname);

    for(i = 0; i< tot; ++i)

    {

        printf(" %s\n", line[i]);

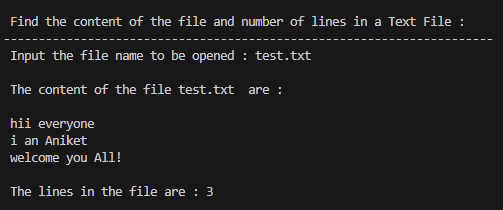
    }

/\*---------------------------------------------------------------------\*/

    printf("\n The lines in the file are : %d\n",tot-1);

    printf("\n");

}



1. Write a program in C to count the number of words and character in a file.

#include <stdio.h>

#include <stdlib.h>

void main()

{

FILE \*fptr;

char ch;

int wrd=1,charctr=1;

char fname[20];

printf("\n\n Count the number of words and characters in a file :\n");

printf("---------------------------------------------------------\n");

printf(" Input the filename to be opened : ");

scanf("%s",fname);

fptr=fopen(fname,"r");

if(fptr==NULL)

{

printf(" File does not exist or can not be opened.");

}

else

{

ch=fgetc(fptr);

printf(" The content of the file %s are : ",fname);

while(ch!=EOF)

{

printf("%c",ch);

if(ch==' '||ch=='\n')

{

wrd++;

}

else

{

charctr++;

}

ch=fgetc(fptr);

}

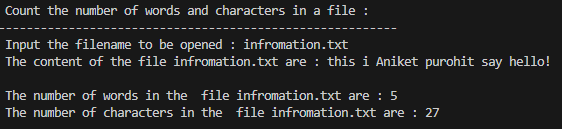
printf("\n The number of words in the file %s are : %d\n",fname,wrd-2);

printf(" The number of characters in the file %s are : %d\n\n",fname,charctr-1);

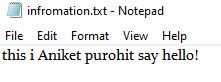
}

fclose(fptr);

}



**“information.txt”**

****

1. Write a program in C to delete a specific line from a file.

#include <stdio.h>

#include <string.h>

#define MAX 256

int main()

{

int lno, ctr = 0;

char ch;

FILE \*fptr1, \*fptr2;

char fname[MAX];

char str[MAX], temp[] = "temp.txt";

printf("\n\n Delete a specific line from a file :\n");

printf("-----------------------------------------\n");

printf(" Input the file name to be opened : ");

scanf("%s",fname);

fptr1 = fopen(fname, "r");

if (!fptr1)

{

printf(" File not found or unable to open the input file!!\n");

return 0;

}

fptr2 = fopen(temp, "w"); // open the temporary file in write mode

if (!fptr2)

{

printf("Unable to open a temporary file to write!!\n");

fclose(fptr1);

return 0;

}

printf(" Input the line you want to remove : ");

scanf("%d", &lno);

lno++;

// copy all contents to the temporary file except the specific line

while (!feof(fptr1))

{

strcpy(str, "\0");

fgets(str, MAX, fptr1);

if (!feof(fptr1))

{

ctr++;

/\* skip the line at given line number \*/

if (ctr != lno)

{

fprintf(fptr2, "%s", str);

}

}

}

fclose(fptr1);

fclose(fptr2);

remove(fname); // remove the original file

rename(temp, fname); // rename the temporary file to original name

fptr1=fopen(fname,"r");

ch=fgetc(fptr1);

printf(" Now the content of the file %s is : \n",fname);

while(ch!=EOF)

{

printf("%c",ch);

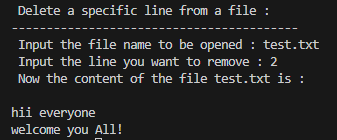
ch=fgetc(fptr1);

}

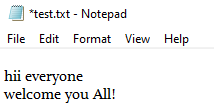
fclose(fptr1);

return 0;

}



**“test.txt”**

****

1. Write a program in C to replace a specific line with another text in a file.

#include <stdio.h>

#include <string.h>

#define MAX 256

int main()

{

FILE \*fptr1, \*fptr2;

int lno, linectr = 0;

char str[MAX],fname[MAX];

char newln[MAX], temp[] = "temp.txt";

printf("\n\n Replace a specific line in a text file with a new text :\n");

printf("-------------------------------------------------------------\n");

printf(" Input the file name to be opened : ");

fgets(fname, MAX, stdin);

fname[strlen(fname) - 1] = '\0';

fptr1 = fopen(fname, "r");

if (!fptr1)

{

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

return 0;

}

fptr2 = fopen(temp, "w");

if (!fptr2)

{

printf("Unable to open a temporary file to write!!\n");

fclose(fptr1);

return 0;

}

/\* get the new line from the user \*/

printf(" Input the content of the new line : ");

fgets(newln, MAX, stdin);

/\* get the line number to delete the specific line \*/

printf(" Input the line no you want to replace : ");

scanf("%d", &lno);

lno++;

// copy all contents to the temporary file other except specific line

while (!feof(fptr1))

{

strcpy(str, "\0");

fgets(str, MAX, fptr1);

if (!feof(fptr1))

{

linectr++;

if (linectr != lno)

{

fprintf(fptr2, "%s", str);

}

else

{

fprintf(fptr2, "%s", newln);

}

}

}

fclose(fptr1);

fclose(fptr2);

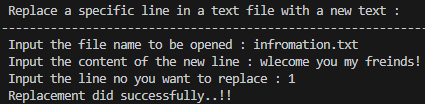
remove(fname);

rename(temp, fname);

printf(" Replacement did successfully..!! \n");

return 0;

}



**“infromation.txt”**

1. Write a program in C to append multiple lines to the end of a file.

#include <stdio.h>

int main ()

{

FILE \* fptr;

int i,n;

char str[100];

char fname[20];

char str1;

printf("\n\n Append multiple lines at the end of a text file :\n");

printf("------------------------------------------------------\n");

printf(" Input the file name to be opened : ");

scanf("%s",fname);

fptr = fopen(fname, "a");

printf(" Input the number of lines to be written : ");

scanf("%d", &n);

printf(" The lines are : \n");

for(i = 0; i< n+1;i++)

{

fgets(str, sizeof(str), stdin);

fputs(str, fptr);

}

fclose (fptr);

//----- Read the file after appended -------

fptr = fopen (fname, "r");

printf("\n The content of the file %s is :\n",fname);

str1 = fgetc(fptr);

while (str1 != EOF)

{

printf ("%c", str1);

str1 = fgetc(fptr);

}

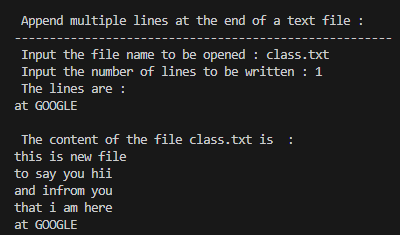
printf("\n\n");

fclose (fptr);

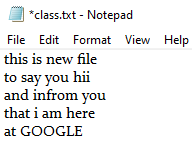
//------- End of reading ------------------

return 0;

}



**“class.txt”**

****

1. Write a program in C to copy a file to another name.

#include <stdio.h>

#include <stdlib.h>

void main()

{

    FILE \*fptr1, \*fptr2;

    charch, fname1[20], fname2[20];

    printf("\n\n Copy a file in another name :\n");

    printf("----------------------------------\n");

    printf(" Input the source file name : ");

    scanf("%s",fname1);

    fptr1=fopen(fname1, "r");

    if(fptr1==NULL)

    {

        printf(" File does not found or error in opening.!!");

        exit(1);

    }

    printf(" Input the new file name : ");

    scanf("%s",fname2);

    fptr2=fopen(fname2, "w");

    if(fptr2==NULL)

    {

        printf(" File does not found or error in opening.!!");

        fclose(fptr1);

        exit(2);

    }

    while(1)

    {

        ch=fgetc(fptr1);

        if(ch==EOF)

        {

            break;

        }

        else

        {

            fputc(ch, fptr2);

        }

    }

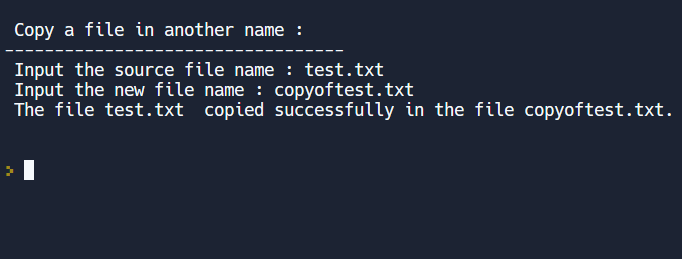
    printf(" The file %s  copied successfully in the file %s. \n\n",fname1,fname2);

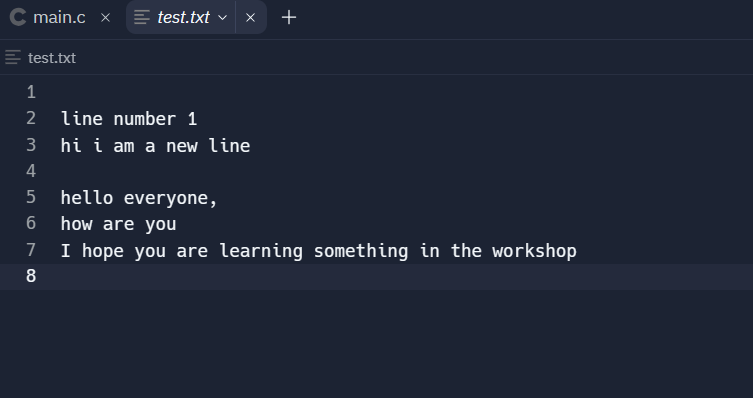
    fclose(fptr1);

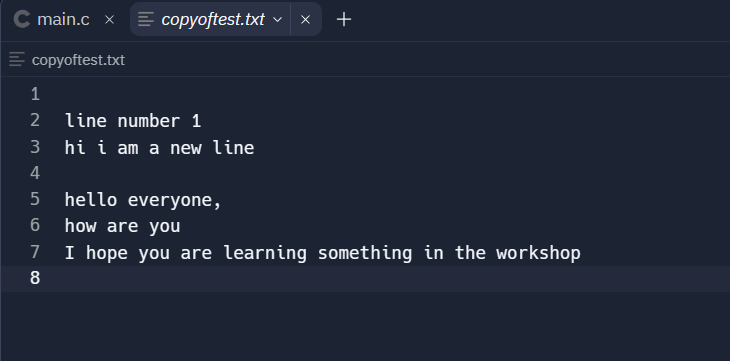
    fclose(fptr2);

    getchar();

}







1. Write a program in C to merge two files and write them to another file.

#include <stdio.h>

#include <stdlib.h>

void main()

{

    FILE \*fold1, \*fold2, \*fnew;

    charch, fname1[20], fname2[20], fname3[30];

    printf("\n\n Merge two files and write it in a new file :\n");

    printf("-------------------------------------------------\n");

    printf(" Input the 1st file name : ");

    scanf("%s",fname1);

    printf(" Input the 2nd file name : ");

    scanf("%s",fname2);

    printf(" Input the new file name where to merge the above two files : ");

    scanf("%s",fname3);

    fold1=fopen(fname1, "r");

    fold2=fopen(fname2, "r");

    if(fold1==NULL || fold2==NULL)

    {

//      perror("Error Message ");

        printf(" File does not exist or error in opening...!!\n");

        exit(EXIT\_FAILURE);

    }

    fnew=fopen(fname3, "w");

    if(fnew==NULL)

    {

//      perror("Error Message ");

        printf(" File does not exist or error in opening...!!\n");

        exit(EXIT\_FAILURE);

    }

    while((ch=fgetc(fold1))!=EOF)

    {

        fputc(ch, fnew);

    }

    while((ch=fgetc(fold2))!=EOF)

    {

        fputc(ch, fnew);

    }

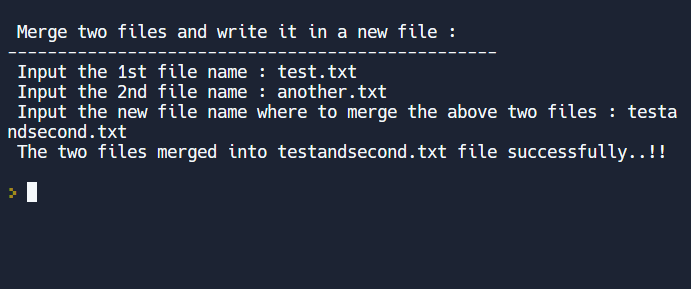
    printf(" The two files merged into %s file successfully..!!\n\n", fname3);

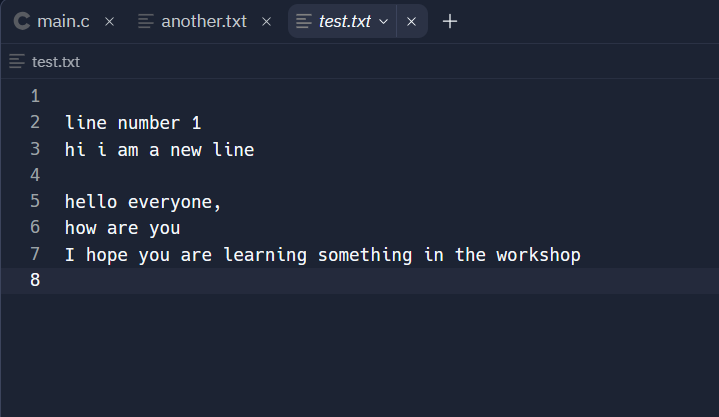
    fclose(fold1);

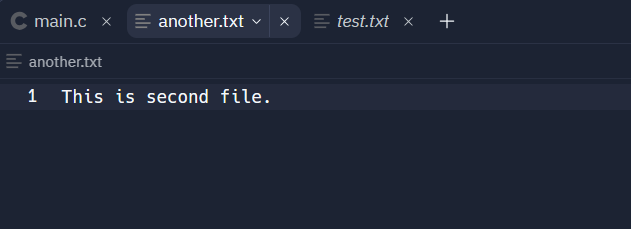
    fclose(fold2);

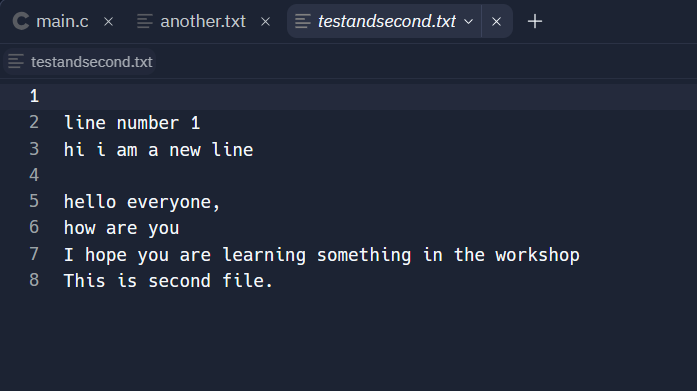
    fclose(fnew);

}









1. C program to print right angle triangle in a pyramid star pattern.

#include <stdio.h>

int main() {

   inti, j, rows;

   printf("Enter the number of rows: ");

   scanf("%d", &rows);

   for (i = 1; i<= rows; ++i) {

      for (j = 1; j <= i; ++j) {

         printf("\* ");

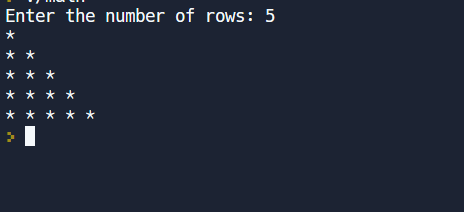
      }

      printf("\n");

   }

   return0;

}



1. C program to print rhombus in a star pattern.

#include <stdio.h>

int main()

{

    inti, j, rows;

    printf("Enter rows: ");

    scanf("%d", &rows);

    for(i=1; i<=rows; i++)

    {

        for(j=1; j<=rows - i; j++)

        {

            printf(" ");

        }

        for(j=1; j<=rows; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

    return0;

}



1. C program to print hollow square and star pattern.

#include <stdio.h>

int main()

{

    inti, j, rows;

    printf("Enter rows: ");

    scanf("%d", &rows);

    for(i=1; i<=rows; i++)

    {

        for(j=1; j<=rows - i; j++)

        {

            printf(" ");

        }

        for(j=1; j<=rows; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

    return0;

}



1. C program for character/alphabet pattern.

A

B C

C D E

D E F G

E F G H I

#include<stdio.h>

void main()

{

    inti,j,k,n;

    printf("Enter the no of lines\n");

    scanf("%d",&n);

    for(i=1;i<=n;i++)

    {

        k = i;

        for(j=1;j<=i;j++,k++)

        {

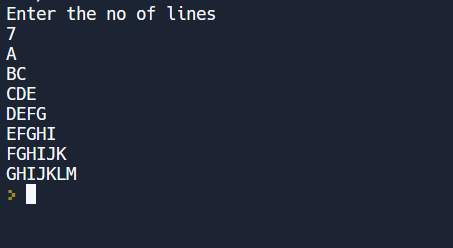
            printf("%c",(char)(k+64));

        }

         printf("\n");

    }

}



1. See a program for triangle number pattern.

1

2 3

3 4 5

4 5 6 7

5 6 7 8 9

#include<stdio.h>

void main()

{

    inti,j,k,n;

    printf("Enter the no of lines\n");

    scanf("%d",&n);

    for(i=1;i<=n;i++)

    {

        k = i;

        for(j=1;j<=i;j++,k++)

        {

            printf("%d",k);

        }

         printf("\n");

    }

}

