

PRACTICAL FILE
OF
“Introduction to Programming Methodology using C ”
(13060112)



SGT UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

Submitted to:

Ms. Simmi Madaan

Assistant professor, CSE

FEAT

Submitted by

Name: Aman

Reg. no.:241306050

BCA 1st Sem (AI/ML)

INDEX

S.No	Name of the Program	Date	Page No.	Signature
1	Program to Convert Celsius to Fahrenheit	16/09/2024	1	
2	Program to Find the Greatest Number Among Three Numbers	16/09/2024	2	
3	Program to Determine the Size of Data Types in C	23/09/2024	3	
4	Program to Check Whether a Character is a Vowel or Consonant	23/09/2024	4	
5	Program to Display an Inverse Triangle Pattern	23/09/2024	5	
6	Program to Display a Pyramid Pattern	30/09/2024	6	
7	Program to Calculate the Average Marks of 50 Students	30/09/2024	7-8	
8	Program to Find the Factorial of a Number Using a Function	30/09/2024	9-10	
9	Program to Count Characters in a String	07/09/2024	11	
10	Program to Find the Smallest Value Among Three Numbers	07/10/2024	12	
11	Program to Calculate Compound Interest	07/10/2024	13	
12	Program to Find the Area of a Circle	14/10/2024	14	
13	Program to Check if a Year is a Leap Year	14/10/2024	15	
14	Program to Calculate Simple Interest	14/10/2024	16	
15	Program to Display Sizes of Different Data Types	28/10/2024	17	

16	Program to Find the Sum of Natural Numbers up to N	28/10/2024	18	
17	Program to Print Multiplication Tables up to N	04/11/2024	19	
18	Program to Swap Two Numbers Using Call by Value and Call by Reference	11/11/2024	20	
19	Program to Sort an Array in Ascending Order Using Bubble Sort	11/11/2024	21	
20	Program to Perform Matrix Addition for 2x2 Arrays	25/11/2024	22	
21	Program to Find the Transpose of a Matrix for 2x2 Arrays	25/11/2024	23	
22	Program to Calculate Factorial Using Recursion	02/12/2024	24	
23	Program to Print Fibonacci Series	02/12/2024	25	
24	Program to Perform Matrix Multiplication for 2x2 Arrays	02/12/2024	26-27	
25	Menu-driven program for string operations Basic operations	16/12/2024	28-30	
26	Program to perform Array of Pointers Displaying a List of Fruits	16/12/2024	31	
27	Program to Implement Pointer to Array Example Accessing an Integer Array	16/12/2024	32	
28	Program to Implement Binary Search	23/12/2024	33	
29	Program to Perform Insertion in an Array	23/12/2024	34	
30	Program to Implement Linear Search	23/12/2024	35	

Program 1

Write a program to convert temperature from Celsius to Fahrenheit by taking input from the user.

```
#include<stdio.h>

int main() {
float c, f;

printf("Enter the Celsius degree: ");

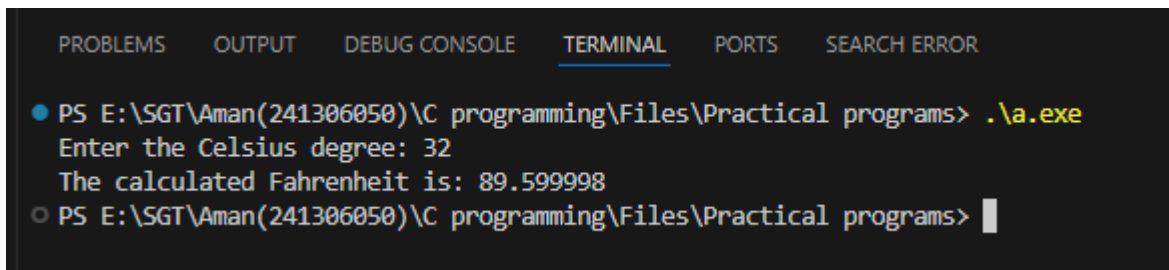
scanf("%f", &c);

f = (c * 9.0 / 5.0) + 32.0;

printf("The calculated Fahrenheit is: %f\n", f);

return 0;
}
```

OUTPUT



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
  Enter the Celsius degree: 32
  The calculated Fahrenheit is: 89.599998
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 2

Write a program to find the greatest number among 3 numbers given by the user.

```
#include <stdio.h>

int main() {
    int a, b, c;
    printf("Enter the value of a: ");
    scanf("%d", &a);
    printf("Enter the value of b: ");
    scanf("%d", &b);
    printf("Enter the value of c: ");
    scanf("%d", &c);
    if (a >= b && a >= c) {
        printf("The greatest number is: %d\n", a);
    }
    else if (b >= a && b >= c) {
        printf("The greatest number is: %d\n", b);
    } else {
        printf("The greatest number is: %d\n", c);
    }
    return 0;
}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter the value of a: 2
Enter the value of b: 3
Enter the value of c: 6
The greatest number is: 6
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 3

C Program to Find the Size of int, float, double and char

```
#include <stdio.h>

int main() {

    printf("Size of int: %zu bytes\n", sizeof(int));

    printf("Size of float: %zu bytes\n", sizeof(float));

    printf("Size of double: %zu bytes\n", sizeof(double));

    printf("Size of char: %zu bytes\n", sizeof(char));

    return 0;
}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
  Size of int: 4 bytes
  Size of float: 4 bytes
  Size of double: 8 bytes
  Size of char: 1 bytes
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 4

C Program to Check Whether a Character is a Vowel or Consonant.

```
#include <stdio.h>

int main() {
    char ch;
    printf("Enter a character: ");
    scanf(" %c", &ch);

    if (isalpha(ch)) {

        char lowerCh = tolower(ch);
        if (lowerCh == 'a' || lowerCh == 'e' || lowerCh == 'i' || lowerCh == 'o' || lowerCh == 'u') {
            printf("%c is a vowel.\n", ch);
        } else {
            printf("%c is a consonant.\n", ch);
        }
    }
}
```

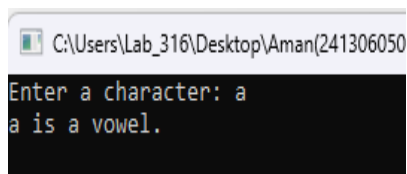
```

    } else {
        printf("%c is not an alphabetic character.\n", ch);
    }
    return 0;
}

```

Output .

Case-1

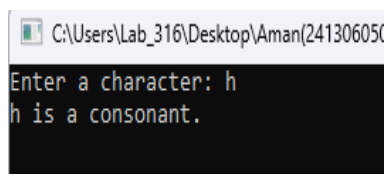


```

C:\Users\Lab_316\Desktop\Aman(241306050)
Enter a character: a
a is a vowel.

```

Case-2

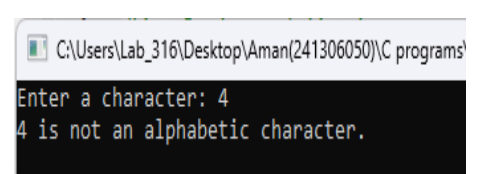


```

C:\Users\Lab_316\Desktop\Aman(241306050)
Enter a character: h
h is a consonant.

```

Case-3



```

C:\Users\Lab_316\Desktop\Aman(241306050)\C programs\
Enter a character: 4
4 is not an alphabetic character.

```

Program 5

Write a program to display inverse triangle pattern.

```

#include <stdio.h>

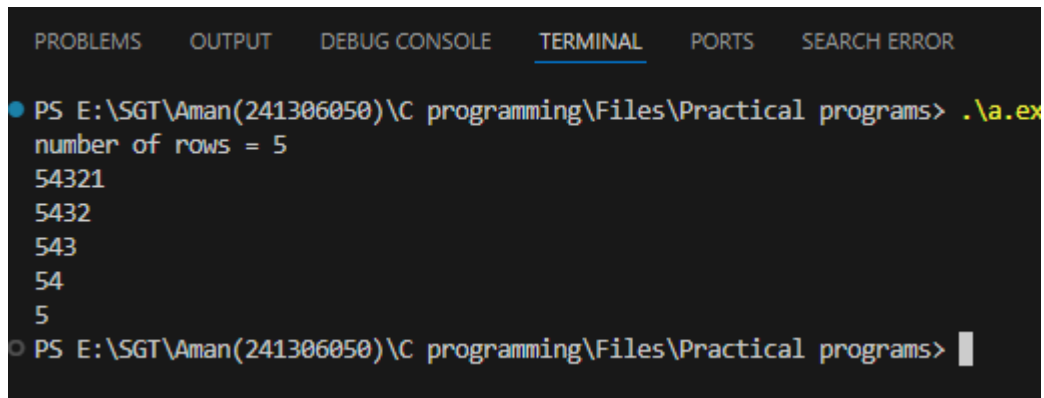
int main(){
    int rows,i,j;
    printf("number of rows = ");
    scanf("%d",&rows);
    for(i=1;i<=rows;i++)

    {
        then j will print numbers from 5 to 1*/
        for(j=5;j>=i;j--)
        { printf("%d",j);
          printf("\n");
        }
    }
    return 0;
}

```


}

Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.ex
number of rows = 5
54321
5432
543
54
5
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> █
```

Program 6

Write a program to display pyramid pattern.

```
#include <stdio.h>
int main()
{
    int i,j,k,rows;
    printf("enter the rows = ");
    scanf("%d",&rows);
    /* i is for the number of rows i=1 to i=rows*/
    for(i=1;i<=rows;i++){

        /* j is to maintain the space from the left side
        so that it starts printing from the middle*/

        for(j=rows;j>=i;j--){
            printf(" ");
        }/*k is to print the value inti piramid form it'll work as
        if i =2, k<=2*2-1 i.e. k<=3 so it'll print three "*" in second row*/
        for(k=1;k<=2*i-1;k++){
            printf("*");
        }
    }
}
```

```

        printf("\n");
    }
    return 0;
}

```

Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
enter the rows = 5
*
***
*****
*****
*****

```

Program 7

Write a program to input marks of 50 students using an array and display and display the average marks.

```

#include <stdio.h>
int main(){
    int i,sum=0,avg;
    int array[50];

    for(i=0;i<50;i++){
        printf("Enter the marks of student %d ",i+1);
        scanf("%d",&array[i]);
    }
    for(i=0;i<50;i++){
        sum+=array[i];
        avg = sum/50;
    }

    printf("\nsum of total 50 students maths marks is :- %d\n",sum);
    printf("avg marks are:- %d ",avg);
}

```

```
return 0;  
}
```

OUTPUT

```
C:\Users\Lab_316\Desktop\An X + v
Enter the marks of student 1 55
Enter the marks of student 2 55
Enter the marks of student 3 55
Enter the marks of student 4 55
Enter the marks of student 5 55
Enter the marks of student 6 52
Enter the marks of student 7 42
Enter the marks of student 8 25
Enter the marks of student 9 44
Enter the marks of student 10 90
Enter the marks of student 11 95
Enter the marks of student 12 98
Enter the marks of student 13 96
Enter the marks of student 14 96
Enter the marks of student 15 65
Enter the marks of student 16 85
Enter the marks of student 17 68
Enter the marks of student 18 95
Enter the marks of student 19 63
Enter the marks of student 20 68
Enter the marks of student 21 78
Enter the marks of student 22 88
Enter the marks of student 23 44
Enter the marks of student 24 55
Enter the marks of student 25 36
Enter the marks of student 26 69
Enter the marks of student 27 68
Enter the marks of student 28 68
Enter the marks of student 29 69
Enter the marks of student 30 66
Enter the marks of student 31 85
Enter the marks of student 32 85
Enter the marks of student 33 99
Enter the marks of student 34 92
Enter the marks of student 35 36
Enter the marks of student 36 66
Enter the marks of student 37 99
Enter the marks of student 38 95
Enter the marks of student 39 100
Enter the marks of student 40 15
Enter the marks of student 41 30
Enter the marks of student 42 30
Enter the marks of student 43 35
Enter the marks of student 44 25
Enter the marks of student 45 65
Enter the marks of student 46 66
Enter the marks of student 47 69
Enter the marks of student 48 65
Enter the marks of student 49 36
Enter the marks of student 50 66

sum of total 50 students maths marks is :- 3257
avg marks are:- 65
```

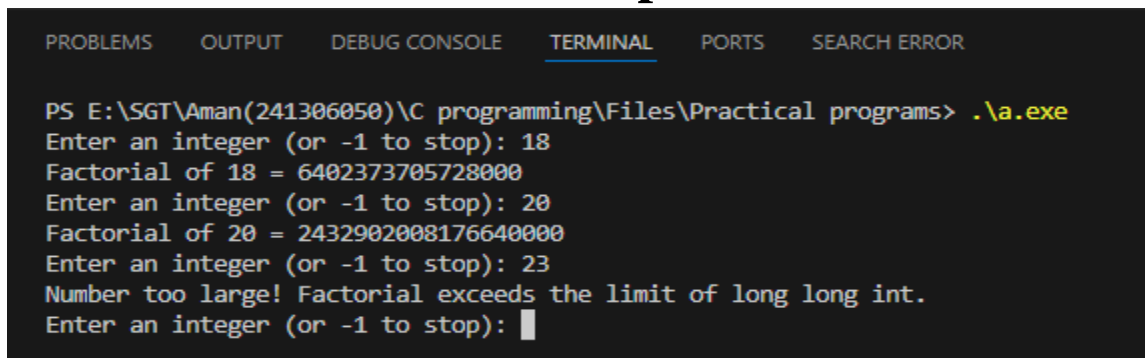
Program 8

write a program to take number users and print its factorial.

```
#include <stdio.h>
int main() {
    int num, i;
    long long int factorial;
    while (1) {
        printf("Enter an integer (or -1 to stop): ");
        if (scanf("%d", &num) != 1) {
            printf("Invalid input. Please enter a valid integer.\n");
            while (getchar() != '\n');
            continue;
        }

        if (num == -1) {
            printf("Program terminated.\n");
            break;
        }
        if (num < 0) {
            printf("Factorial is not defined for negative numbers.\n");
        } else if (num > 20) {
            printf("Number too large! Factorial exceeds the limit of long long int.\n");
        } else {
            factorial = 1;
            for (i = 1; i <= num; ++i) {
                factorial *= i;
            }
            printf("Factorial of %d = %lld\n", num, factorial);
        }
    }
    return 0; }
```

Output

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active and underlined), 'PORTS', and 'SEARCH ERROR'. The terminal shows the execution of a program named 'a.exe'. The user enters '18', and the program outputs 'Factorial of 18 = 6402373705728000'. The user enters '20', and the program outputs 'Factorial of 20 = 2432902008176640000'. The user enters '23', and the program outputs 'Number too large! Factorial exceeds the limit of long long int.'. The user then enters an empty line, and the program prompts 'Enter an integer (or -1 to stop):' with a cursor.

Using Function of Factorial

```

#include <stdio.h>
long long int factorial(int num) {
    long long int fact = 1;
    int i;
    for (i = 1; i <= num; ++i) {
        fact *= i;
    }
    return fact;
}
int main() {
    int num;
    while (1) {
        printf("Enter an integer (or -1 to stop): ");
        if (scanf("%d", &num) != 1) {
            printf("Invalid input. Please enter a valid integer.\n");
            while (getchar() != '\n');
            continue;
        }

        if (num == -1) {
            printf("Program terminated.\n");
            break;
        }
        if (num < 0) {
            printf("Factorial is not defined for negative numbers.\n");
        } else if (num > 20) {
            printf("Number too large! Factorial exceeds the limit of long long int.\n");
        } else {
            printf("Factorial of %d = %lld\n", num, factorial(num));
        }
    }
    return 0;
}

```

Output

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter an integer (or -1 to stop): 18
Factorial of 18 = 6402373705728000
Enter an integer (or -1 to stop): 20
Factorial of 20 = 2432902008176640000
Enter an integer (or -1 to stop): 23
Number too large! Factorial exceeds the limit of long long int.
Enter an integer (or -1 to stop): █

```

Program 9

String Characters Count

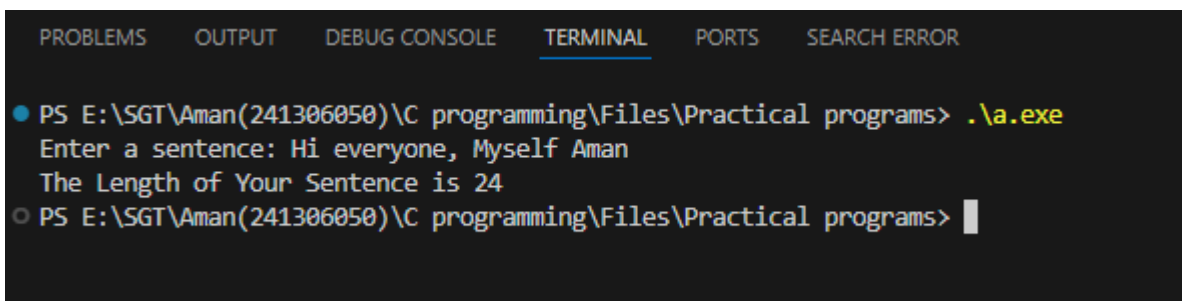
```
#include <stdio.h>
#include <string.h>

int main()
{
    char str[100];
    int len;

    printf("Enter a sentence: ");

    fgets(str, sizeof(str), stdin);
    str[strcspn(str, "\n")] = '\0';
    len = strlen(str);
    printf("The Length of Your Sentence is %d\n", len);
    return 0;
}
```

OUTPUT



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
  Enter a sentence: Hi everyone, Myself Aman
  The Length of Your Sentence is 24
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

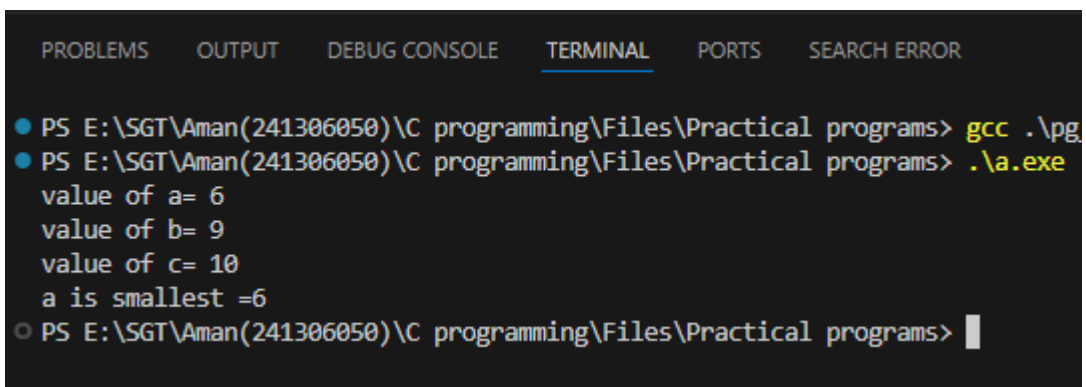
Program 10

Write a Program to find the smallest value from three values

```
#include <stdio.h>

int main()
{
    int a,b,c;
    printf("value of a= ");
    scanf("%d",&a);
    printf("value of b= ");
    scanf("%d",&b);
    printf("value of c= ");
    scanf("%d",&c);
    if ((a<b)&&(a<c))
    { printf("a is smallest =%d",a);
    }
    else if ((b<a)&&(b<c))
    { printf("b is smallest=%d",b);
    }
    else
    { printf("c is smallest =%d",c);}
    return 0;
}
```

Output



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. Below the tabs, the terminal shows the following commands and output:

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> gcc .\pg
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
value of a= 6
value of b= 9
value of c= 10
a is smallest =6
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```


Program 11

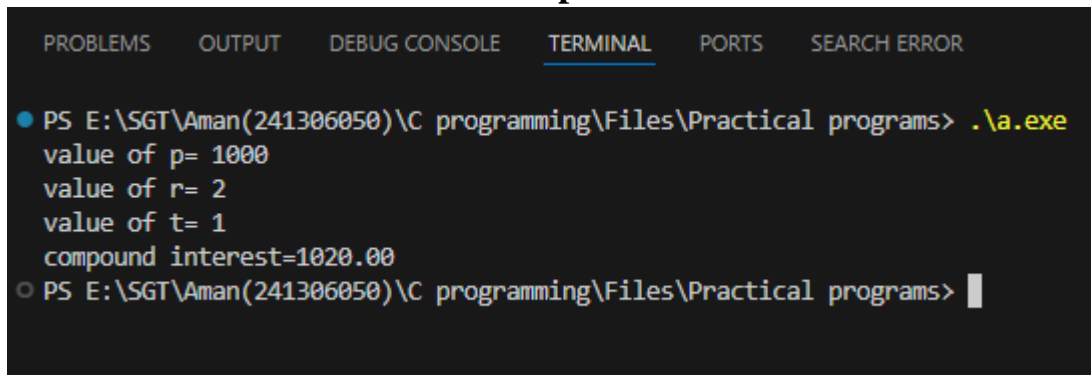
Write a Program to find the compound interest:

```
#include <stdio.h>
#include <math.h>
int main()
{
    float p,r,t,CI;
    printf("value of p= ");
    scanf("%f",&p);
    printf("value of r= ");
    scanf("%f",&r);
    printf("value of t= ");
    scanf("%f",&t);

    CI=p*pow(1+r/100,t);
    printf("compound interest=%.2f",CI);

    return 0;
}
```

Output

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. The terminal shows the execution of a program. The prompt is 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>'. The user enters '.\a.exe'. The program outputs: 'value of p= 1000', 'value of r= 2', 'value of t= 1', and 'compound interest=1020.00'. The prompt returns, and the user enters a blank line.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
value of p= 1000
value of r= 2
value of t= 1
compound interest=1020.00
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> 
```

Program 12

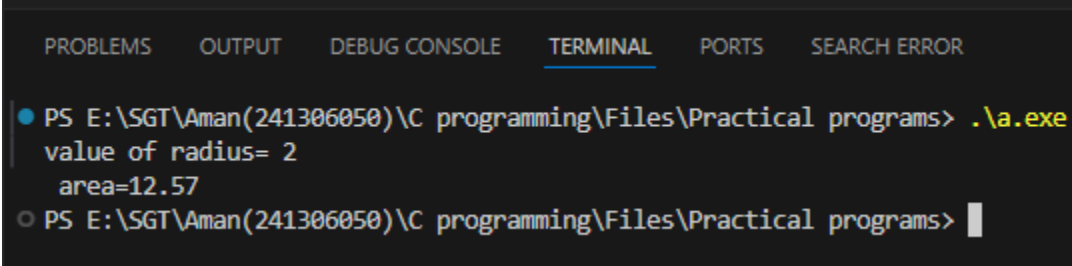
Write a Program to find the area of the circle.

```
#include <stdio.h>
#include <math.h>
int main()
{
    float radius , area;
    printf("value of radius= ");
    scanf("%f",&radius);

    area=acos(-1) *radius *radius;
    printf(" area=%.2f",area);

    return 0;
}
```

Output



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. The terminal content shows a command prompt where the user has run a program named 'a.exe'. The program prompts for the radius, which is entered as '2'. It then calculates and displays the area as '12.57'.

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
value of radius= 2
area=12.57
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 13

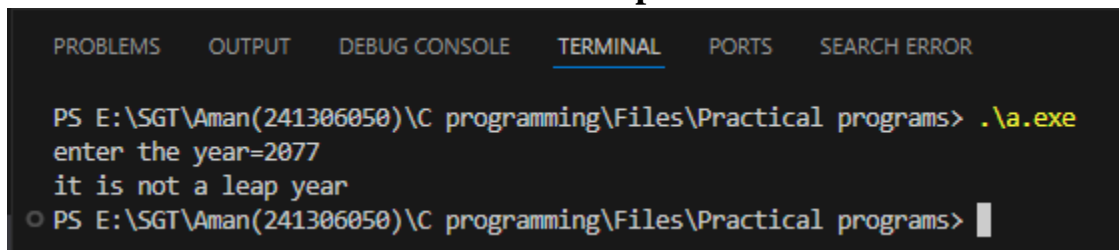
Write a Program to find the leap year

```
#include <stdio.h>

int main() {
    int year;
    printf("enter the year=");
    scanf("%d",&year);
    if (year%4==0)
    {
        printf("it is leap year");
    }
    else
    {
        printf("it is not a leap year");
    }

    return 0;
}
```

Output



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. The terminal content shows the command prompt 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' followed by the command '.\a.exe' in yellow. The program then prompts 'enter the year=' and the user enters '2077'. The program outputs 'it is not a leap year'. The prompt continues with 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' and a cursor.

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
enter the year=2077
it is not a leap year
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> █
```

Program 14

Write a Program to find the simple interest:

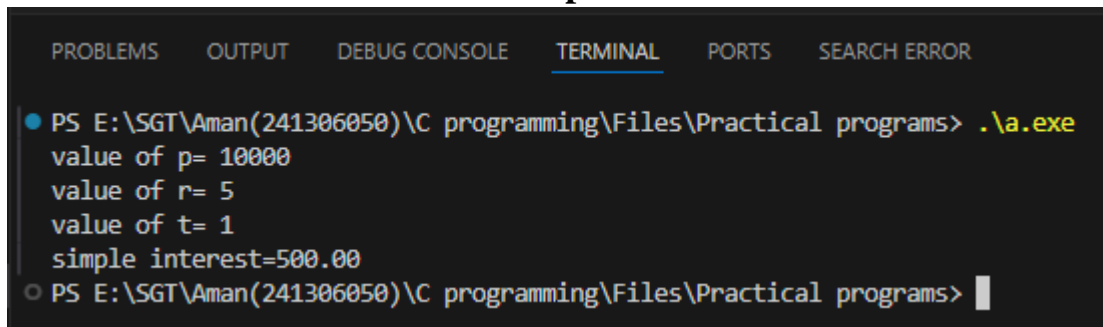
```
#include <stdio.h>

int main()
{
    float p,r,t,SI;
    printf("value of p= ");
    scanf("%f",&p);
    printf("value of r= ");
    scanf("%f",&r);
    printf("value of t= ");
    scanf("%f",&t);

    SI= (p*r*t)/100;
    printf("simple interest=%.2f",SI);

    return 0;
}
```

Output



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. The terminal content shows the execution of a program named '.\a.exe'. The program prompts for three values: 'value of p= 10000', 'value of r= 5', and 'value of t= 1'. It then outputs 'simple interest=500.00'. The prompt 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' is visible at the start and end of the terminal session.

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
value of p= 10000
value of r= 5
value of t= 1
simple interest=500.00
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>
```

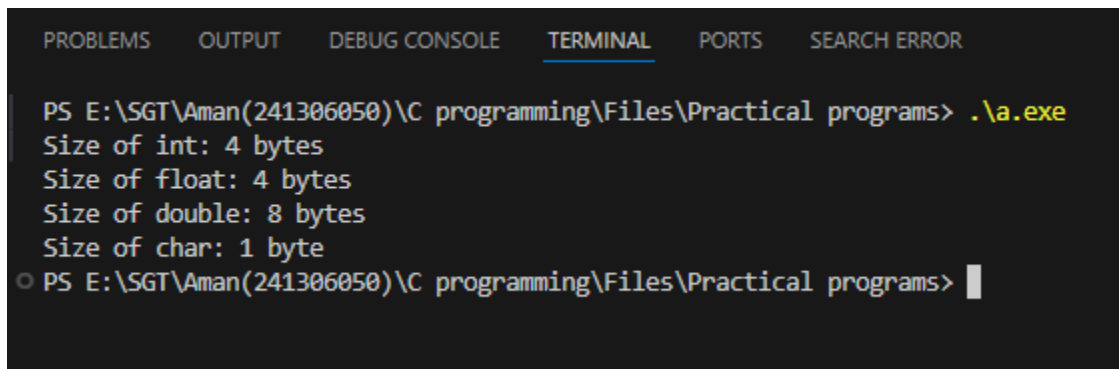
Program 15

Display size of data types

```
#include <stdio.h>

int main() {
    printf("Size of int: %zu bytes\n", sizeof(int));
    printf("Size of float: %zu bytes\n", sizeof(float));
    printf("Size of double: %zu bytes\n", sizeof(double));
    printf("Size of char: %zu byte\n", sizeof(char));
    return 0;
}
```

Output



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), 'PORTS', and 'SEARCH ERROR'. The terminal content shows the command prompt 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' followed by the command '.\a.exe' in yellow. The output of the program is displayed in four lines: 'Size of int: 4 bytes', 'Size of float: 4 bytes', 'Size of double: 8 bytes', and 'Size of char: 1 byte'. Below this, the prompt is repeated with a cursor, indicating the program has finished execution.

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> 
```

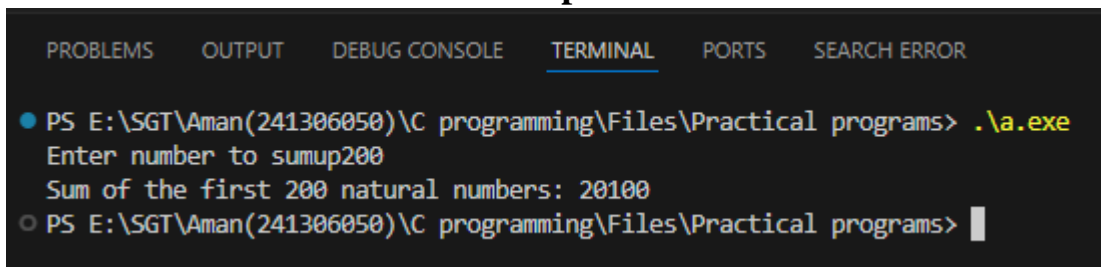
Program 16

Write a Program to print sum of 1 to n^{th} numbers using for loop

```
#include <stdio.h>

int main() {
    int i,j;
    printf("Enter number to sumup");
    scanf("%d",&j);
    int sum = 0;
    for (i = 1; i <= j; i++) {
        sum += i;
    }
    printf("Sum of the first %d natural numbers: %d\n",j, sum);
}
```

Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter number to sumup200
Sum of the first 200 natural numbers: 20100
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

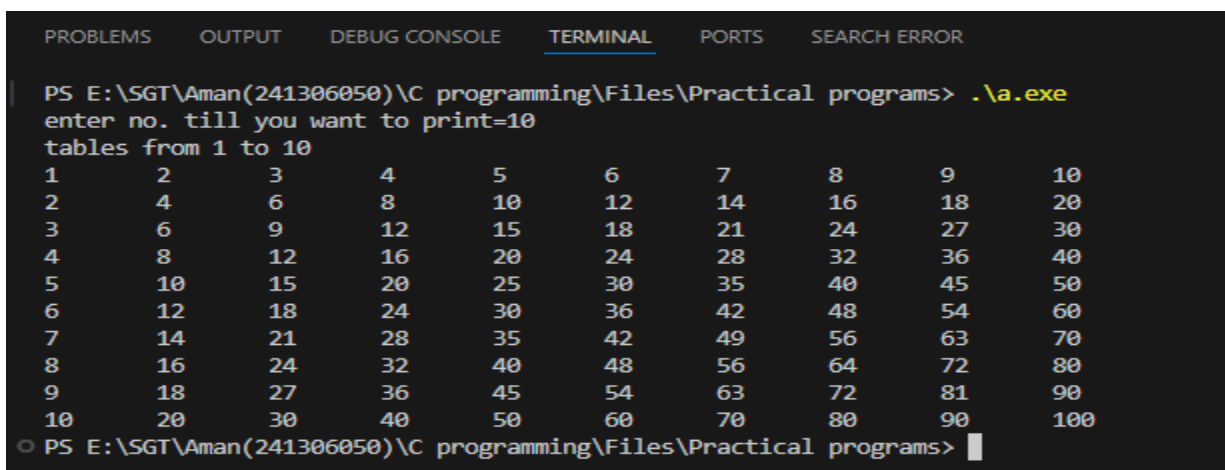
Program 17

Write a Program to print multiplication table of first n natural numbers using for loop in table structure

```
#include<stdio.h>

int main(){
    int i,j,l;
    printf("enter no. till you want to print=");
    scanf("%d",&l);
    printf("tables from 1 to %d\n",l);
    for(j=1;j<=10;j++){
        for(i=1;i<=l;i++)
        {
            printf("%d\t",j*i);
        }
        printf("\n");
    }
    return 0;
}
```

Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

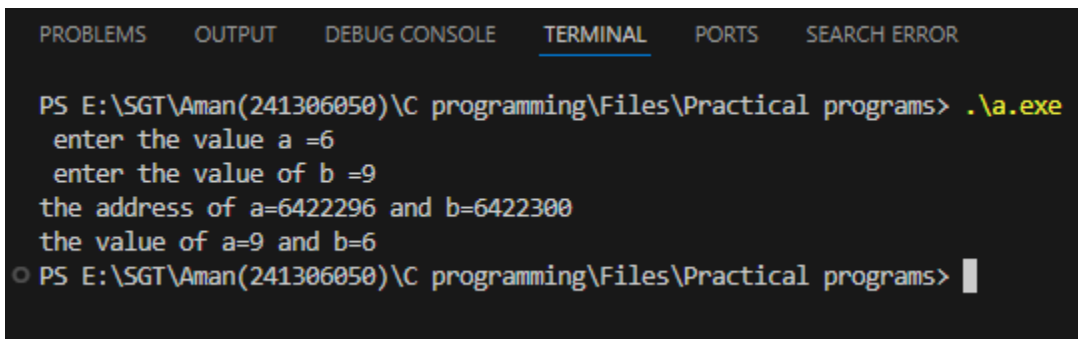
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
enter no. till you want to print=10
tables from 1 to 10
1      2      3      4      5      6      7      8      9      10
2      4      6      8      10     12     14     16     18     20
3      6      9      12     15     18     21     24     27     30
4      8      12     16     20     24     28     32     36     40
5      10     15     20     25     30     35     40     45     50
6      12     18     24     30     36     42     48     54     60
7      14     21     28     35     42     49     56     63     70
8      16     24     32     40     48     56     64     72     80
9      18     27     36     45     54     63     72     81     90
10     20     30     40     50     60     70     80     90     100
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> █
```

Program 18

WRITE A PROGRAM TO PRINT THE ADDRESS AND THE VALUE OF THE VARIABLE USING CALL BY VALUE AND CALL BY REFERENCE

```
int swap(int *x, int *y);
#include <stdio.h>
int main(){
    int a,b,c,d;
    printf(" enter the value a =");
    scanf("%d",&a);
    printf(" enter the value of b =");
    scanf("%d",&b);
    swap(&a,&b);
}
int swap(int *x, int *y){
    int temp;
    temp=x;
    x=y;
    y=temp;
    printf("the address of x=%d and y=%d\n",x,y);
    printf("the value of x=%d and y=%d",*x,*y);
    return 0;
}
```

OUTPUT



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

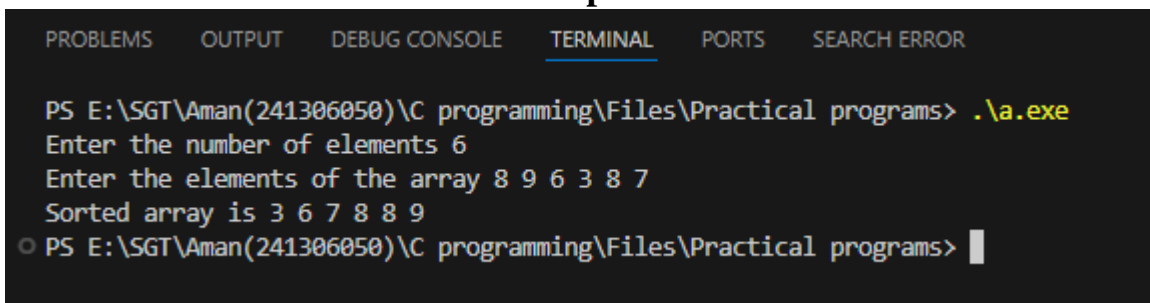
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
    enter the value a =6
    enter the value of b =9
the address of a=6422296 and b=6422300
the value of a=9 and b=6
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> █
```


Program 19

Write a Program to Sorting Array into Ascending Order (Using Bubble Sort)

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter the number of elements ");
    scanf("%d", &n);
    int arr[n];
    int r;
    printf("Enter the elements of the array ");
    for (r=0;r<n;r++)
    {
        scanf("%d", &arr[r]);
    }
    int i, j;
    for (i=0;i<n;i++)
    {
        for (j=0;j<n-i-1;j++)
        {
            if (arr[j] > arr[j+1])
            {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    printf("Sorted array is ");
    for (i=0;i<n;i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

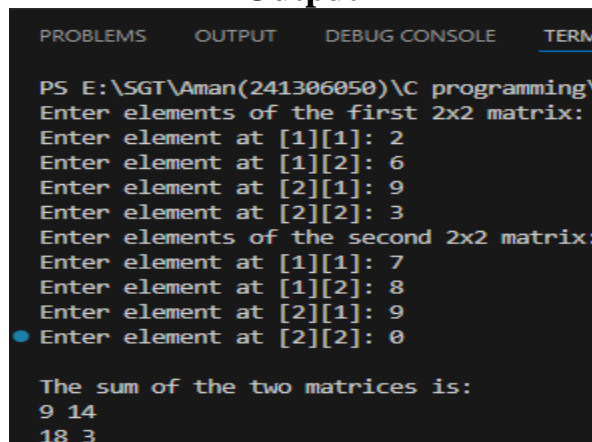
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter the number of elements 6
Enter the elements of the array 8 9 6 3 8 7
Sorted array is 3 6 7 8 8 9
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 20

Write a program to perform matrix addition on 2D Arrays

```
#include <stdio.h>
int main() {
    int i, j;
    int matrix1[2][2], matrix2[2][2], result[2][2];
    printf("Enter elements of the first 2x2 matrix:\n");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            printf("Enter element at [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix1[i][j]);
        }
    }
    printf("Enter elements of the second 2x2 matrix:\n");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            printf("Enter element at [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix2[i][j]);
        }
    }
    // Calculate the sum of the matrices
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
    printf("\nThe sum of the two matrices is:\n");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }
    return 0; }
```

Output



```
PS E:\SGT\Aman(241306050)\C programming>
Enter elements of the first 2x2 matrix:
Enter element at [1][1]: 2
Enter element at [1][2]: 6
Enter element at [2][1]: 9
Enter element at [2][2]: 3
Enter elements of the second 2x2 matrix:
Enter element at [1][1]: 7
Enter element at [1][2]: 8
Enter element at [2][1]: 9
Enter element at [2][2]: 0

The sum of the two matrices is:
9 14
18 3
```

Program 21

Write a program to perform Matrix transpose on 2D Array.

```
#include<stdio.h>

int main() {
    int i,j;
    int matrix[2][2],transpose[2][2];

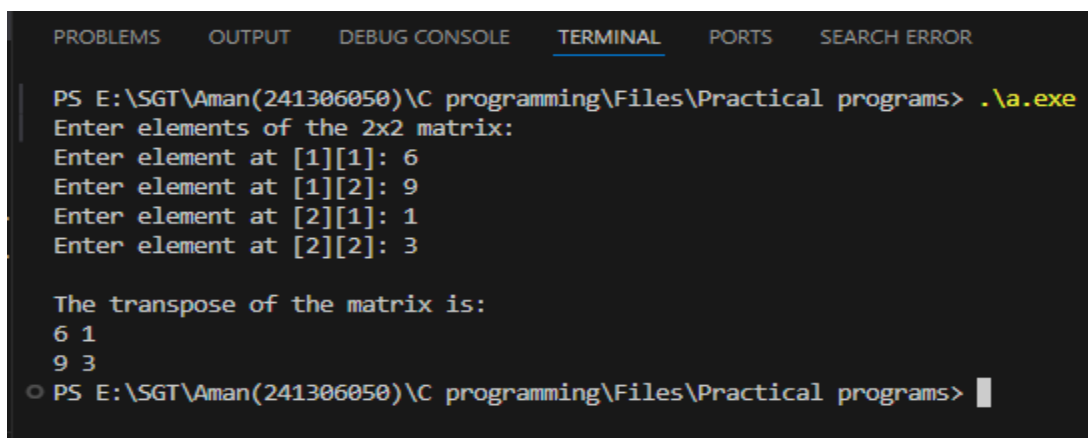
    printf("Enter elements of the 2x2 matrix:\n");    for(i=0;i<2;i++) {        for(j=0;j<2;j++) {
        printf("Enter element at [%d][%d]: ",i+1,j+1);        scanf("%d",&matrix[i][j]);
    }
}

for(i=0;i<2;i++) {        for(j=0;j<2;j++) {
    transpose[j][i]=matrix[i][j];
}
}

printf("\nThe transpose of the matrix is:\n");    for(i=0;i<2;i++) {        for(j=0;j<2;j++) {
    printf("%d ",transpose[i][j]);
}
printf("\n");
}

return 0; }
```

OUTPUT



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'SEARCH ERROR'. The terminal text shows the command prompt 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' followed by the execution of '.\a.exe'. The program prompts the user to 'Enter elements of the 2x2 matrix:' and then asks for four elements: [1][1]: 6, [1][2]: 9, [2][1]: 1, and [2][2]: 3. After the input, it displays 'The transpose of the matrix is:' followed by the transposed matrix: 6 1 and 9 3. The prompt returns to 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>'.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter elements of the 2x2 matrix:
Enter element at [1][1]: 6
Enter element at [1][2]: 9
Enter element at [2][1]: 1
Enter element at [2][2]: 3

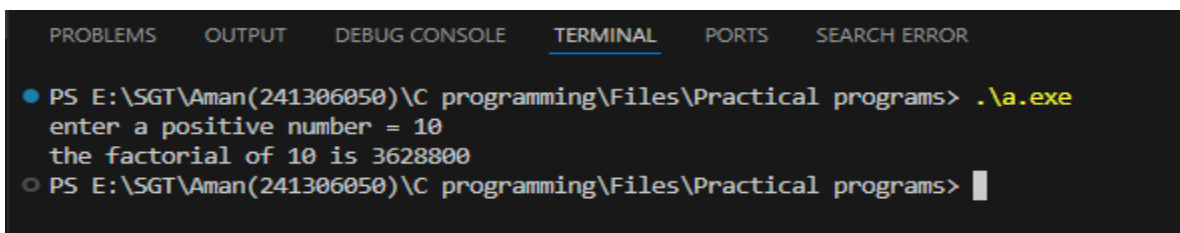
The transpose of the matrix is:
6 1
9 3
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> 
```

Program 22

WRITE A PROGRAM TO FIND THE FACTORIAL USING RECURSION

```
int factorial(int num);
#include<stdio.h>
int main()
{
    int n, num;
    printf("enter a positive number = ");
    scanf("%d",&n);
    if(num<0){
        printf("factorial of negative nums are not defined ");
    }
    else{
        printf("the factorial of %d is %d",n,factorial(n));
    }
    return 0;
}
int factorial(int num) {
    if(num == 0||num == 1){
        return 1;}
    else{
        return num*factorial(num-1);
    }
}
```

Output



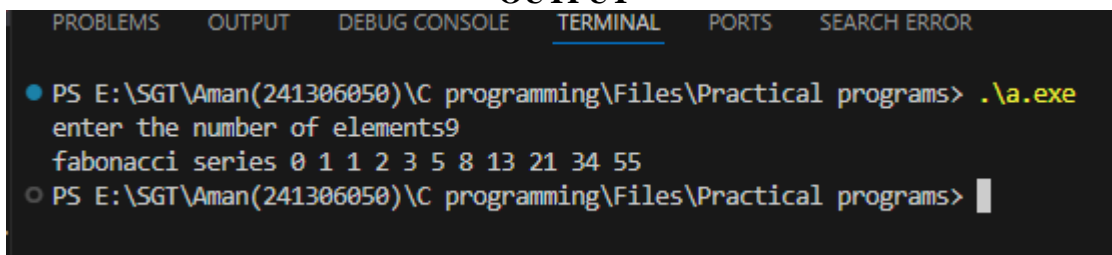
The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'SEARCH ERROR'. Below the tabs, the terminal shows the command prompt 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' followed by the command '.\a.exe'. The output of the program is displayed as 'enter a positive number = 10' and 'the factorial of 10 is 3628800'. The prompt then shows 'PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>' with a cursor.

Program 23

WRITE A PROGRAM TO PRINT THE FIBONACCI SERIES

```
int fibseries(int num);
#include<stdio.h>
int main(){
    int n;
    printf("enter the number of elements");
    scanf("%d",&n);
    printf("fabonacci series ");
    printf("%d %d ",0,1);
    fibseries(n);
    return 0;
}
int fibseries(int n){
    static int n1=0,n2=1,n3;
    if(n>0){
        n3=n1+n2;
        n1=n2;
        n2=n3;
        printf("%d ",n3);
        fibseries(n-1);
    }
    return 0;
}
```

OUTPUT



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), 'PORTS', and 'SEARCH ERROR'. Below the tabs, the terminal shows the following text:

```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
enter the number of elements9
fabonacci series 0 1 1 2 3 5 8 13 21 34 55
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 24
WRITE A PROGRAM TO PERFORM MATRIX MULTIPLICATION IN
2D ARRAY

```
#include <stdio.h>
int main()
{
int A[2][2],B[2][2],C[2][2],i,j,k;
printf("Enter the elements of the first matrix\n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++){
printf("Enter element [%d][%d] ",i+1,j+1);
scanf("%d",&A[i][j]);
}}
printf("Enter the elements of the second matrix\n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++){
printf("Enter element [%d][%d] ",i+1,j+1);
scanf("%d",&B[i][j]);
}}
for(i=0;i<2;i++)
{
for(j=0;j<2;j++){
C[i][j]=0;
for(k=0;k<2;k++){
C[i][j]+=A[i][k]*B[k][j];
}}}
printf("Multiplication of the matrix \n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++){
printf("%d\t",C[i][j]);
}
printf("\n");
}
return 0;}
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

● PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter the elements of the first matrix
Enter element [1][1] 2
Enter element [1][2] 3
Enter element [2][1] 2
Enter element [2][2] 3
Enter the elements of the second matrix
Enter element [1][1] 1
Enter element [1][2] 3
Enter element [2][1] 6
Enter element [2][2] 9
Multiplication of the matrix
20      33
20      33
○ PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> |
```

Program 25

Write a menu driven program to implement the following string operations:

- Calculate length of a string
- Concatenate at the end of a given string
- Copy one string to another
- Compare contents of two strings

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[100], str2[100], str3[100];
    int choice, i;
    do {
        printf("\nMenu:\n");
        printf("1. Calculate length of a string\n");
        printf("2. Concatenate at the end of a given string\n");
        printf("3. Copy one string to another\n");
        printf("4. Compare contents of two strings\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter a string: ");
                scanf("%s", str1);
                for (i = 0; str1[i] != '\0'; i++);
                printf("Length of the string: %d\n", i);
                break;
            case 2:
                printf("Enter the first string: ");
                scanf("%s", str1);
                printf("Enter the second string to concatenate: ");
                scanf("%s", str2);
                for (i = 0; str1[i] != '\0'; i++);
                int j;
                for (j = 0; str2[j] != '\0'; j++, i++) {
                    str1[i] = str2[j];
                }
                str1[i] = '\0';
                printf("Concatenated string: %s\n", str1);
                break;
            case 3:
                printf("Enter a string to copy: ");
                scanf("%s", str1);
```



```

for (i = 0; str1[i] != '\0'; i++) {
    str2[i] = str1[i];
}
str2[i] = '\0';
printf("Copied string: %s\n", str2);
break;
case 4:
printf("Enter the first string: ");
scanf("%s", str1);
printf("Enter the second string: ");
scanf("%s", str2);
for (i = 0; str1[i] != '\0' && str2[i] != '\0'; i++) {
    if (str1[i] != str2[i]) {
        break;
    }
}
if (str1[i] == '\0' && str2[i] == '\0') {
    printf("Strings are equal\n");
} else {
    printf("Strings are not equal\n");
}
break;
case 5:
printf("Exiting program.\n");
break;
default:
printf("Invalid choice. Try again.\n");
}
} while (choice != 5);
return 0;}

```

OUTPUTS

Case 1

```
Menu:
1. Calculate length of a string
2. Concatenate at the end of a given string
3. Copy one string to another
4. Compare contents of two strings
5. Exit
Enter your choice: 1
Enter a string: Aman
Length of the string: 4
```

Case 2

```
Menu:
1. Calculate length of a string
2. Concatenate at the end of a given string
3. Copy one string to another
4. Compare contents of two strings
5. Exit
Enter your choice: 2
Enter the first string: Fave
Enter the second string to concatenate: Aman
Concatenated string: FaveAman
```

Case 3

```
Menu:
1. Calculate length of a string
2. Concatenate at the end of a given string
3. Copy one string to another
4. Compare contents of two strings
5. Exit
Enter your choice: 3
Enter a string to copy: Fave
Copied string: Fave
```

Case 4

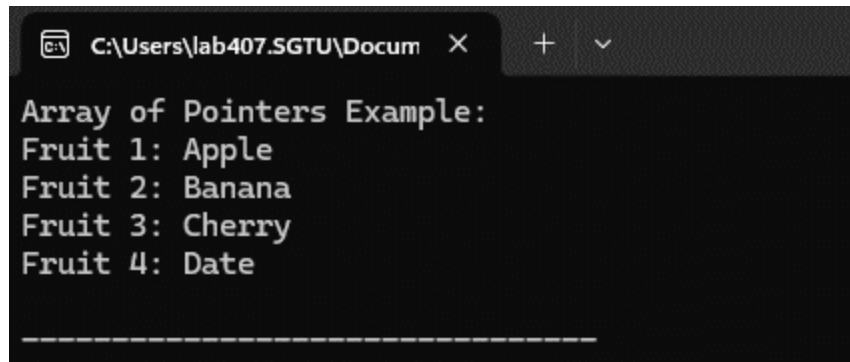
```
Menu:
1. Calculate length of a string
2. Concatenate at the end of a given string
3. Copy one string to another
4. Compare contents of two strings
5. Exit
Enter your choice: 4
Enter the first string: Noob
Enter the second string: fave
Strings are not equal
```

Program 26

Array of Pointers Example Displaying a List of Fruits

```
#include <stdio.h>
int main() {
    const char *fruits[] = {"Apple", "Banana", "Cherry", "Date"};
    int n = sizeof(fruits) / sizeof(fruits[0]);
    printf("Array of Pointers Example:\n");
    for (int i = 0; i < n; i++) {
        printf("Fruit %d: %s\n", i + 1, fruits[i]);
    }
    return 0;
}
```

OUTPUT

A screenshot of a terminal window with a dark background. The window title bar shows the file path "C:\Users\lab407.SGTU\Docum" and standard window controls. The output text is displayed in a light green monospace font. It starts with "Array of Pointers Example:" followed by four lines: "Fruit 1: Apple", "Fruit 2: Banana", "Fruit 3: Cherry", and "Fruit 4: Date". A dashed line is visible at the bottom of the terminal area.

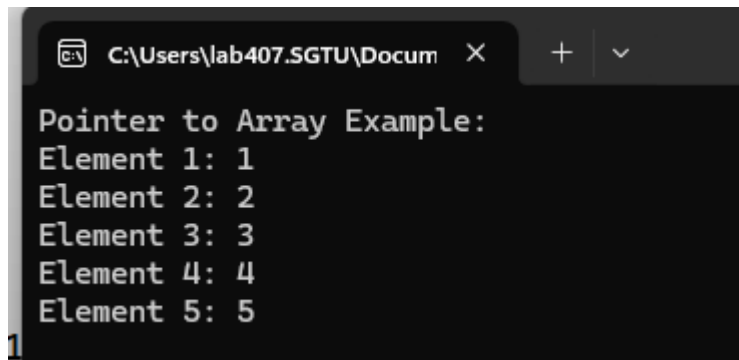
```
C:\Users\lab407.SGTU\Docum  X  +  v
Array of Pointers Example:
Fruit 1: Apple
Fruit 2: Banana
Fruit 3: Cherry
Fruit 4: Date
-----
```

Program 27

Pointer to Array Example Accessing an Integer Array

```
#include <stdio.h>
int main() {
    int numbers[5] = {1, 2, 3, 4, 5};
    int (*ptr)[5] = &numbers;
    printf("Pointer to Array Example:\n");
    for (int i = 0; i < 5; i++) {
        printf("Element %d: %d\n", i + 1, (*ptr)[i]);
    }
    return 0;
}
```

OUTPUT



```
C:\Users\lab407.SGTU\Docum
Pointer to Array Example:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Element 5: 5
```

Program 28

Program to Implement Binary Search

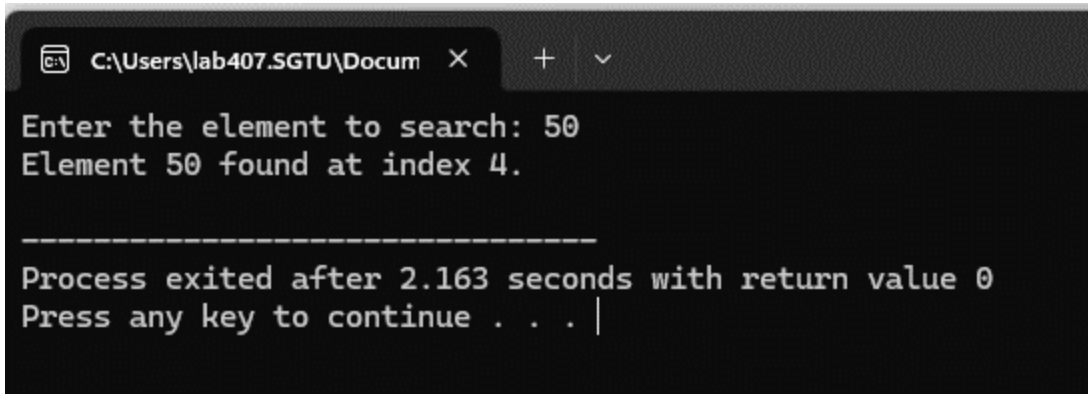
```
#include <stdio.h>
int binarySearch(int arr[], int size, int key) {
    int low = 0, high = size - 1, mid;
    while (low <= high) {
        mid = (low + high) / 2;
        if (arr[mid] == key) {
            return mid;
        } else if (arr[mid] < key) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return -1;
}
int main() {
    int arr[] = {10, 20, 30, 40, 50, 60, 70, 80, 90};
    int size = sizeof(arr) / sizeof(arr[0]);
    int key;

    printf("Enter the element to search: ");
    scanf("%d", &key);

    int result = binarySearch(arr, size, key);

    if (result != -1) {
        printf("Element %d found at index %d.\n", key, result);
    } else {
        printf("Element %d not found in the array.\n", key);
    }
    return 0;}
```

OUTPUT



```
C:\Users\lab407.SGTU\Docum X + v
Enter the element to search: 50
Element 50 found at index 4.

-----
Process exited after 2.163 seconds with return value 0
Press any key to continue . . . |
```

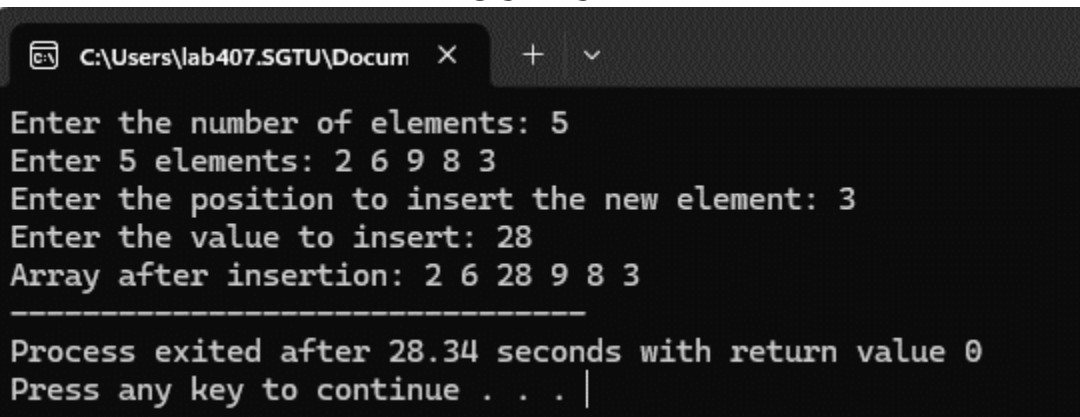
Program 29

Program to Perform Insertion in an Array

```
#include <stdio.h>
int main() {
    int arr[100], n, pos, value;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the position to insert the new element: ");
    scanf("%d", &pos);
    printf("Enter the value to insert: ");
    scanf("%d", &value);
    for (int i = n; i >= pos; i--) {
        arr[i] = arr[i - 1];
    }
    arr[pos - 1] = value;

    printf("Array after insertion: ");
    for (int i = 0; i <= n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

OUTPUT



The screenshot shows a terminal window with a dark background. The title bar at the top indicates the file path 'C:\Users\lab407.SGTU\Docum'. The program's output is displayed in a monospaced font. It prompts the user to enter the number of elements (5), then the elements themselves (2 6 9 8 3). Next, it asks for the position to insert a new element (3) and the value to insert (28). The resulting array is shown as '2 6 28 9 8 3'. A separator line of dashes follows. The program then displays 'Process exited after 28.34 seconds with return value 0' and 'Press any key to continue . . . |'.

```
C:\Users\lab407.SGTU\Docum X + v
Enter the number of elements: 5
Enter 5 elements: 2 6 9 8 3
Enter the position to insert the new element: 3
Enter the value to insert: 28
Array after insertion: 2 6 28 9 8 3
-----
Process exited after 28.34 seconds with return value 0
Press any key to continue . . . |
```

Program 30

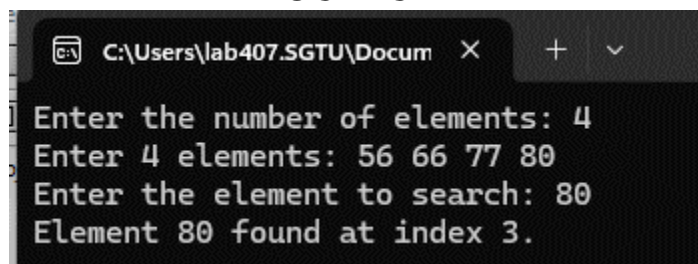
Program to Implement Linear Search

```
#include <stdio.h>
int main() {
    int arr[100], n, key, found = 0;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &key);

    for (int i = 0; i < n; i++) {
        if (arr[i] == key) {
            printf("Element %d found at index %d.\n", key, i);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Element %d not found in the array.\n", key);
    }

    return 0;
}
```

OUTPUT



The screenshot shows a terminal window with the following text:

```
C:\Users\lab407.SGTU\Docum  X  +  v
Enter the number of elements: 4
Enter 4 elements: 56 66 77 80
Enter the element to search: 80
Element 80 found at index 3.
```