JAIN
DEEMED-TO-BE UNIVERSITY

Programming in C

Lab File

Experiments No.: 34 to 64

Subject Code: 16BCA1C05L
Class: I Year I Semester (BCA)

Prepared By
Suman Garai
JU2020BCAS19059

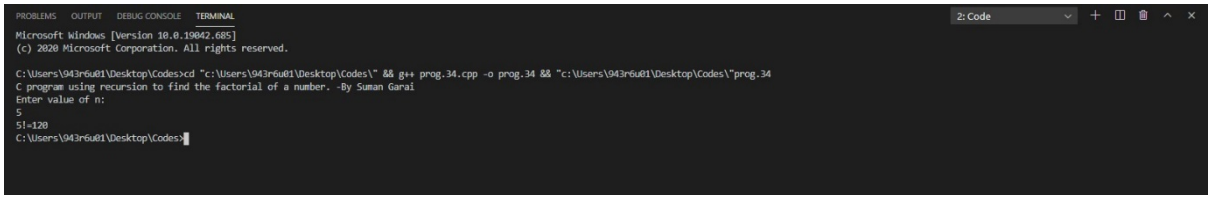
Experiment 34:

C program using recursion to find the factorial of a number.

Code:

```
#include<stdio.h>
main()
{
    int fact(int);
    int n,f;
    printf("C program using recursion to find the
    factorial of a number. -By Suman Garai\n");
    printf("Enter value of n: \n");
    scanf("%d",&n);
    f=fact(n);
    printf("%d!=%d",n,f);
}
int fact(int n)
{
    if(n==1)
    return 1;
    return (n*fact(n-1));
}
```

Output:



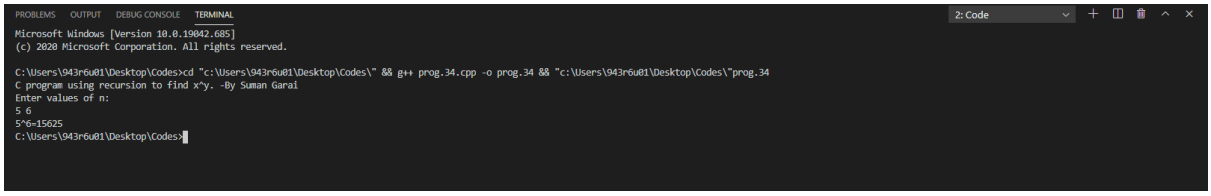
Experiments 35:

C program using recursion to find x^y.

Code:

```
#include<stdio.h>
main()
{
    int power(int,int);
    printf("C program using recursion to find x^y. -
    By Suman Garai\n");
    int x,y,result;
    printf("Enter values of n:\n");
    scanf("%d %d",&x,&y);
    result=power(x,y);
    printf("%d^%d=%d",x,y,result);
}
int power(int x,int y)
{
    if(y==1)
        return x;
    return (x*power(x,--y));
}
```

Output:



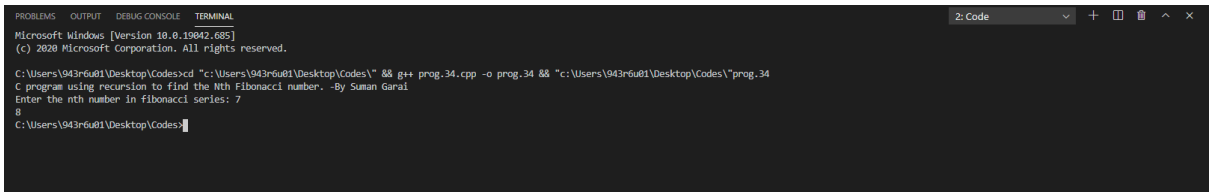
Experiment 36:

C program using recursion to find the nth Fibonacci number.

Code:

```
#include<stdio.h>
main()
{
 int fibo(int);
 int n,result;
 printf("C program using recursion to find the Nth
 Fibonacci number. -By Suman Garai\n");
 printf("Enter the nth number in fibonacci series:
");  scanf("%d", &n);
 result = fibo(n);
 printf("%d",result);
}
int fibo(int n)
{
 if (n == 1)
 {
 return 0;
 }
 else if (n == 2)
 {
 return 1;
 }
 else
 {
 return(fibo(n - 1) + fibo(n - 2));
 }
} n result;
}
```

Output:



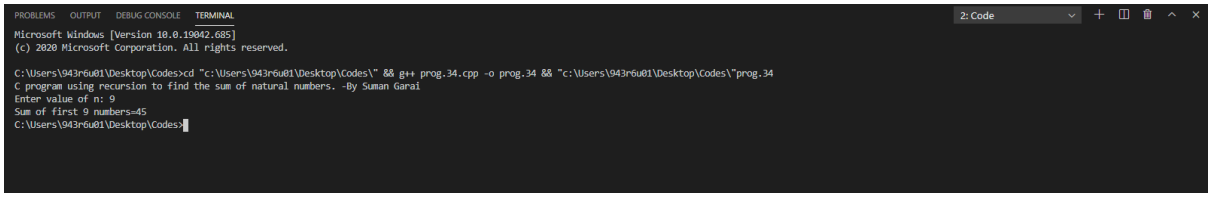
Experiment 37:

C program using recursion to find the sum of natural numbers.

Code:

```
#include<stdio.h>
main()
{
    int sum(int);
    int n,s;
    printf("C program using recursion to find the sum
    of natural numbers. -By Suman Garai\n");
    printf("Enter value of n: ");
    scanf("%d",&n);
    s=sum(n);
    printf("Sum of first %d numbers=%d",n,s);
}
int sum(int n)
{
    if(n==1)
    return 1;
    return (n+sum(n-1));
}
```

Output:



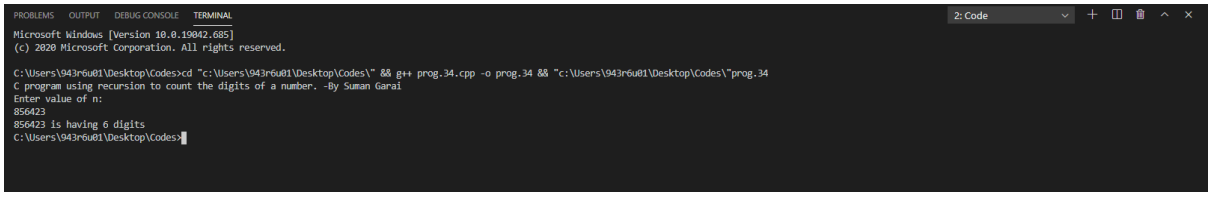
Experiment 38:

C program using recursion to count the digits of a number.

Code:

```
#include<stdio.h>
main()
{
    int count(int);
    int n,result;
    printf("C program using recursion to count the
    digits of a number. -By Suman Garai\n");
    printf("Enter value of n: \n");
    scanf("%d",&n);
    result=count(n);
    printf("%d is having %d digits",n,result);
}
int count(int n)
{
    if(n<=0)
    return 0;
    return (1+count(n/10));
}
```

Output:



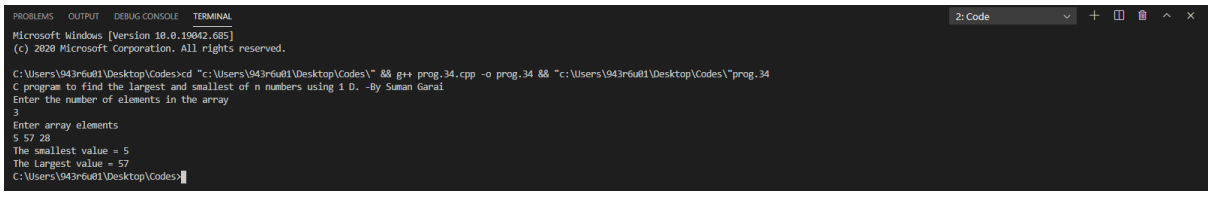
Experiment 39:

C program to find the largest and smallest of n numbers using 1 D arrays.

Code:

```
#include<stdio.h>
main()
{
    int n,a[20],min,max,i;
    printf("C program to find the largest and
    smallest of n numbers using 1 D. -By Suman
    Garai\n");
    printf("Enter the number of elements in the
    array\n");
    scanf("%d",&n);
    printf("Enter array elements\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    min=max=a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]>max)
            max=a[i];
        if(a[i]<min)
            min=a[i];
    }
    printf("The smallest value = %d",min);
    printf("\nThe Largest value = %d",max);
}
```

Output:



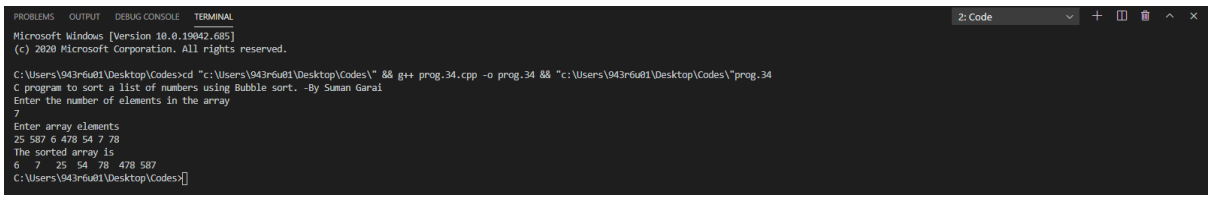
Experiment 40:

C program to sort a list of numbers using Bubble sort.

Code:

```
#include<stdio.h>
main()
{
    int n,a[20],temp,j,i;
    printf("C program to sort a list of numbers using
    Bubble sort. -By Suman Garai\n");
    printf("Enter the number of elements in the
    array\n");
    scanf("%d",&n);
    printf("Enter array elements\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    printf("The sorted array is\n");
    for(i=0;i<n;i++)
        printf("%-4d",a[i]);
}
```

Output:



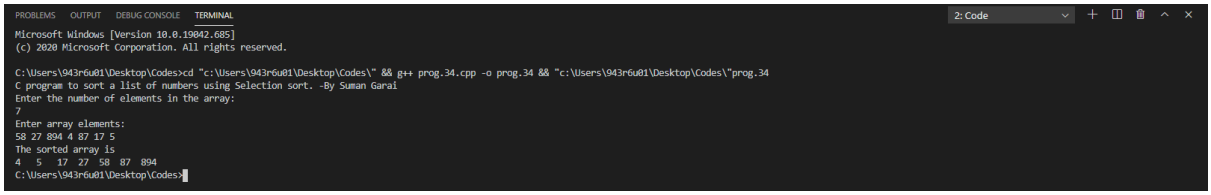
Experiment 41:

C program to sort a list of numbers using Selection sort.

Code:

```
#include<stdio.h>
main()
{
    int n,a[20],temp,j,i;
    printf("C program to sort a list of numbers using
    Selection sort. -By Suman Garai\n");
    printf("Enter the number of elements in the
    array:\n"); scanf("%d",&n);
    printf("Enter array elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0; i< n-1 ; i++)
    {
        for(j= i+1 ; j<n ; j++)
        {
            if ( a[j] < a[i])
            {
                temp=a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("The sorted array is\n");
    for(i=0;i<n;i++)
        printf("%-4d",a[i]);
}
```

Output:



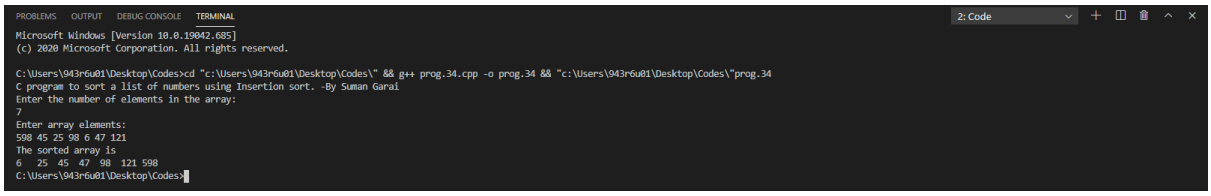
Experiment 42:

C program to sort a list of numbers using Insertion sort.

Code:

```
#include<stdio.h>
main()
{
    int n,a[20],temp,j,i;
    printf("C program to sort a list of numbers using
    Insertion sort. -By Suman Garai\n");
    printf("Enter the number of elements in the
    array: \n");
    scanf("%d",&n);
    printf("Enter array elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=1;i<n;i++){
        temp=a[i];
        j=i-1;
        while((temp<a[j])&&(j>=0)){
            a[j+1]=a[j];
            j=j-1;
        }
        a[j+1]=temp;
    }
    printf("The sorted array is\n");
    for(i=0;i<n;i++)
        printf("%-4d",a[i]);
}
```

Output:



Experiment 43:

C program to find search for a given number using Linear search.

Code:

```
#include<stdio.h>
main()
{
    int n,a[20],num,j,i;
    printf("C program to find search for a given
    number using Linear search. -By Suman Garai\n");
    printf("Enter the number of elements in the
    array\n");
    scanf("%d",&n);
    printf("Enter array elements\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    printf("Enter the number to be searched\n");
    scanf("%d",&num);
    for(i=0;i<n;i++){
        if(a[i]==num)
        {
         printf("%d found at index %d",num,i);
         return 1;
        }
    }
    printf("%d is not present in the given
array\n",num);
}
```

Output:

Experiment 44:

C program to find search for a given number using Binary search.

Code:

```
#include<stdio.h>
main()
{
    int i, first, last, middle, n, search, arr[100];
    printf("C program to find search for a given
    number using Binary search. -By Suman Garai\n");
    printf("Enter number of elements:\n");
    scanf("%d",&n);
    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d",&arr[i]);
    printf("Enter the value to find:\n");
    scanf("%d", &search);
    first = 0;
    last = n - 1;
    middle = (first+last)/2;
    while (first <= last)
    {
        if (arr[middle] < search)
            first = middle + 1;
        else if (arr[middle] == search) {
             printf("%d is present at position
             %d.\n", search, middle+1);  break; }
        else {
            last = middle - 1;
            middle = (first + last)/2; }
    }
    if (first > last)
        printf("Not found! %d is not present in the
list.\n", search);
}
```

Output:

Experiment 45:

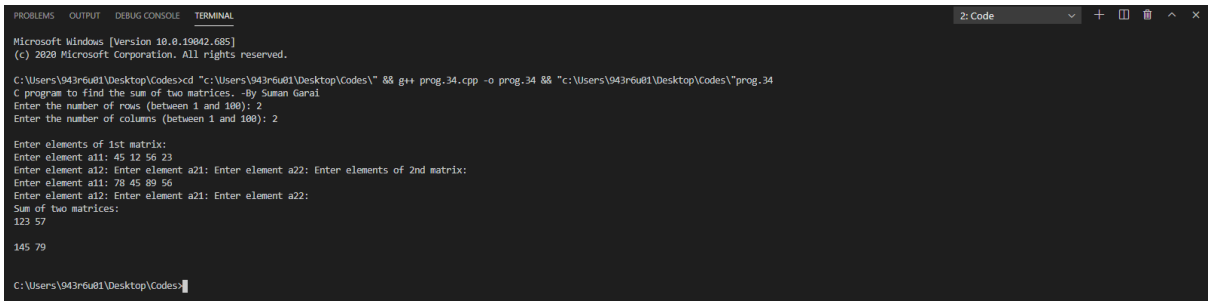
C program to find the sum of two matrices.

Code:

```
#include <stdio.h>
int main()
{
    int r, c, a[100][100], b[100][100],
    sum[100][100], i, j;
    printf("C program to find the sum of two
    matrices. -By Suman Garai\n");
    printf("Enter the number of rows (between 1 and
    100): ");  scanf("%d", &r);
    printf("Enter the number of columns (between 1
    and 100): ");  scanf("%d", &c);
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            printf("Enter element a%d%d: ", i + 1, j
            + 1);
            scanf("%d", &a[i][j]);
        }
    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            printf("Enter element a%d%d: ", i + 1, j
            + 1);
            scanf("%d", &b[i][j]);
        }
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            sum[i][j] = a[i][j] + b[i][j];
        }
    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
```

```
            {
                printf("%d ", sum[i][j]);
                if (j == c - 1)
                    printf("\n\n");
            }
    return 0;
}
```

Output:



Experiment 46:

C program to find the product of two matrices.

Code:

```
#include<stdio.h>
main()
{
    int
    i,j,k,r1,c1,r2,c2,a[10][10],b[10][10],c[10][10];
    printf("C program to find the product of two
    matrices. -By Suman Garai\n");
    printf("Enter the no. of rows and columns of
    matrix 1 : "); scanf("%d %d",&r1,&c1);
    printf("Enter the no. of rows and columns of
    matrix 2 : "); scanf("%d %d",&r2,&c2);
    if(c1==r2)
    {
        printf("Enter the matrix 1\n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
            {
                scanf("%d",&a[i][j]);
            }
        }
        printf("Enter the matrix 2\n");
        for(i=0;i<r2;i++)
        {
            for(j=0;j<c2;j++)
            {
                scanf("%d",&b[i][j]);
            }
        }
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                c[i][j]=0;
                for(k=0;k<r1;k++)
```

```
                {
                    c[i][j]=c[i][j]+a[i][k]*b[k][j];
                }
            }
        }
        printf("Product of matrices\n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                printf("%d\t",c[i][j]);
            }
            printf("\n");
        }
    }
    else
        printf("Matrix multiplication not possible");
printf("\n");
}
```

Output:

Experiment 47:

C program to transpose a given matrix.

Code:

```
#include <stdio.h>
main()
{
    int a[10][10], transpose[10][10], r, c, i, j;
    printf("C program to transpose a given matrix. -BY
    Suman Garai\n");
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    printf("\nEnter matrix elements:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            printf("Enter element a%d%d: ", i + 1, j +
            1);  scanf("%d", &a[i][j]);
        }
    printf("\nEntered matrix: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            printf("%d ", a[i][j]);
            if (j == c - 1)
                printf("\n");
        }
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
        {
            transpose[j][i] = a[i][j];
        }
    printf("\nTranspose of the matrix:\n");
    for (i = 0; i < c; ++i)
        for (j = 0; j < r; ++j)
        {
            printf("%d ", transpose[i][j]);
            if (j == r - 1)
                printf("\n");
        }
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
2: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to transpose a given matrix. -BY Suman Garai
Enter rows and columns: 2 2
Enter matrix elements:
Enter element a11: 78 56 12 30
Enter element a12: Enter element a21: Enter element a22:
Entered matrix:
78 56
12 30
Transpose of the matrix:
78 12
56 30
C:\Users\943r6u01\Desktop\Codes>

Experiment 48:

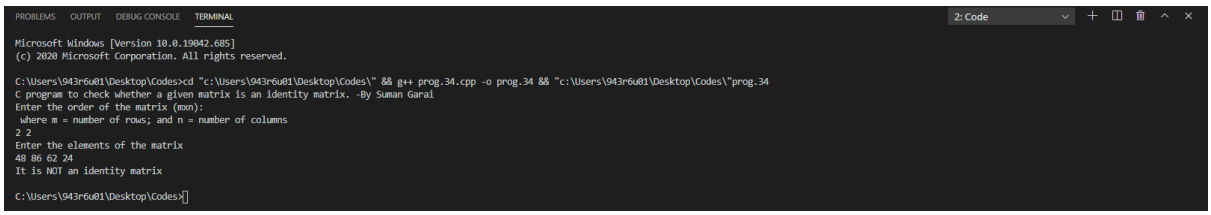
C program to check whether a given matrix is an identity matrix.

Code:

```
#include <stdio.h>
int main (void)
{
    int a[10][10];
    int i = 0, j = 0, row = 0, col = 0;
    int flag = 0;
    printf ("C program to check whether a given
    matrix is an identity matrix. -By Suman
    Garai\n");
    printf ("Enter the order of the matrix
    (mxn):\n");
    printf (" where m = number of rows; and");
    printf (" n = number of columns\n");
    scanf ("%d %d", &row, &col);
    if(row!=col)
    {
        printf("Not an idetity matrix : Rows and
        column is not equal\n"); return 0;
    }
    printf ("Enter the elements of the matrix\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            scanf ("%d", &a[i][j]);
        }
    }
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            if (i == j && a[i][j] != 1)
            {
                flag = -1;
                break;
```

```
            }
            else if (i != j && a[i][j] != 0)
            {
                flag = -1;
                break;
            }
        }
    }
    if (flag == 0)
        printf ("It is a IDENTITY MATRIX\n");
    else
        printf ("It is NOT an identity matrix\n");
    return 0;
}
```

Output:



Experiment 49:

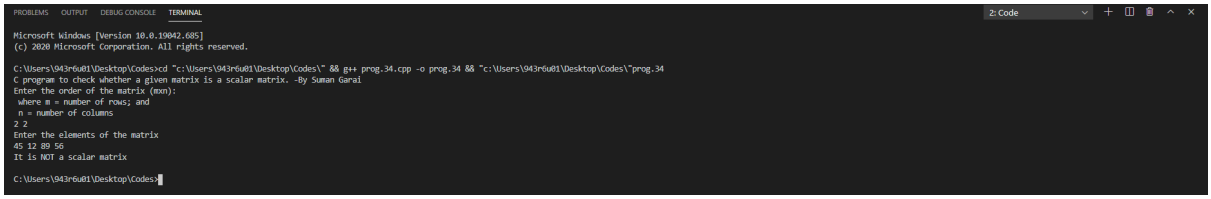
C program to check whether a given matrix is a scalar matrix.

Code:

```
#include <stdio.h>
main ()
{
    int a[10][10];
    int i = 0, j = 0, row = 0, col = 0,scalar;
    int flag = 0;
    printf("C program to check whether a given matrix
    is a scalar matrix. -By Suman Garai\n");
    printf ("Enter the order of the matrix mxn):\n");
    printf (" where m = number of rows; and\n");
    printf (" n = number of columns\n");
    scanf ("%d %d", &row, &col);
    if(row!=col)
    {
        printf("Not an idetity matrix : Rows and
        column is not equal\n"); return 0;
    }
    printf ("Enter the elements of the matrix\n");
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            scanf ("%d", &a[i][j]);
        }
    }
    scalar=a[0][0];
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            if (i == j && a[i][j] != scalar)
            {
                flag = -1;
                break;
            }
```

```
            else if (i != j && a[i][j] != 0)
            {
                flag = -1;
                break;
            }
        }
    }
    if (flag == 0)
    {
        printf ("It is a scalar MATRIX\n");
    }
    else
    {
        printf ("It is N0T a scalar matrix\n");
    }
}
```

Output:



Experiment 50:

C program to find the frequency of a character in a string.

Code:

```
#include<stdio.h>
#include <string.h>
main()
{
    char s[50],ch;
    int i,count=0;
    printf("C program to find the frequency of a
    character in a string. -By Suman Garai\n");
    printf("Enter the String: ");
    gets(s);
    printf("Enter the character\n");
    ch=getchar();
    for(i=0;s[i]!='\0';i++)
        if(ch==s[i])
        count++;
    printf("%c is present %d times\n",ch,count);
}
```

Output:

Experiment 51:

C program to reverse a string.

Code:

```
#include<stdio.h>
#include <string.h>
main()
{
    char s[50],rev[50];
    int i,length=0,j=0;
    printf("C program to reverse a string. -By Suman
    Garai\n");
    printf("Enter the String: ");
    gets(s);
    for(i=0;s[i]!='\0';i++)
    length++;
    for(i=length-1;i>=0;i--)
    {
        rev[j]=s[i];
        j++;
    }
    rev[j]='\0';
    printf("The reversed string is: ");
    puts(rev);
}
```

Output:

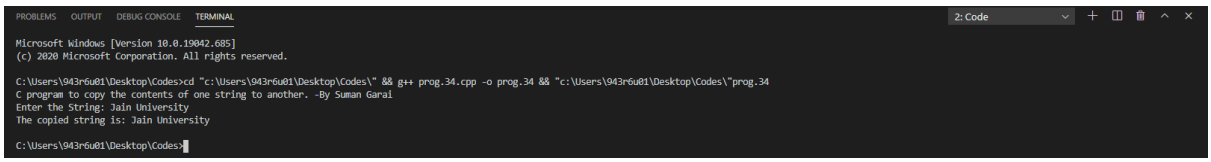
Experiment 52:

C program to copy the contents of one string to another.

Code:

```
#include<stdio.h>
#include <string.h>
main()
{
    char s[50],rev[50];
    int i;
    printf("C program to copy the contents of one
    string to another. -By Suman Garai\n");
    printf("Enter the String: ");
    gets(s);
    for(i=0;s[i]!='\0';i++)
        rev[i]=s[i];
    rev[i]='\0';
    printf("The copied string is: ");
    puts(rev);
}
```

Output:



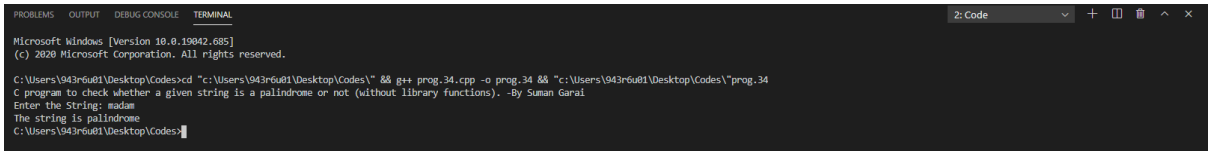
Experiment 53:

C program to check whether a given string is a palindrome or not (without library functions).

Code:

```
#include<stdio.h>
#include <string.h>
int main()
{
    char s[50],rev[50];
    int i,length=0,j=0;
    printf("C program to check whether a given string
is a palindrome or not (without library functions). -
By Suman Garai\n");
    printf("Enter the String: ");
    gets(s);
    for(i=0;s[i]!='\0';i++)
        length++;
    length--;
    for(i=0;i<(length/2);i++)
    {
        if(s[i]!=s[length])
        {
            printf("String is not palindrome\n");
            return 0;
        }
        length--;
    }
    printf("The string is palindrome");
    return 0;
}
```

Output:



Experiment 54:

C program to remove all blank spaces and punctuation symbols from a string.

Code:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
main()
{
    char s[50];
    int i,len=0,j;
    printf(" C program to remove all blank spaces and
    punctuation symbols from a string. -By Suman
    Garai\n");
    printf("Enter the String: ");
    gets(s);
    for(i=0;s[i]!='\0';i++)
        len++;
    for(i=0;i<len;i++)
    {
        if(ispunct(s[i]))
        {
             printf("Removing Punctuation '%c'
             ...\n",s[i] );
            for(j=i;j<len;j++)
                s[j]=s[j+1];
            len--;
        }
        if(s[i]==' ')
        {
            printf("Removing Space ' ' ...\n");
            for(j=i;j<len;j++)
                s[j]=s[j+1];
            len--;
        }
    }
    printf("Output: \n");
    puts(s);
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
2: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to remove all blank spaces and punctuation symbols from a string. -By Suman Garai
Enter the String: afdknmadkfmn. kafmniaf;s a;dal;
Removing Punctuation '.' ...
Removing Space ' ' ...
Removing Punctuation ';' ...
Removing Space ' ' ...
Removing Punctuation ';' ...
Removing Punctuation ';' ...
Output:
afdknmadkfmnkafmniafsadal
C:\Users\943r6u01\Desktop\Codes>

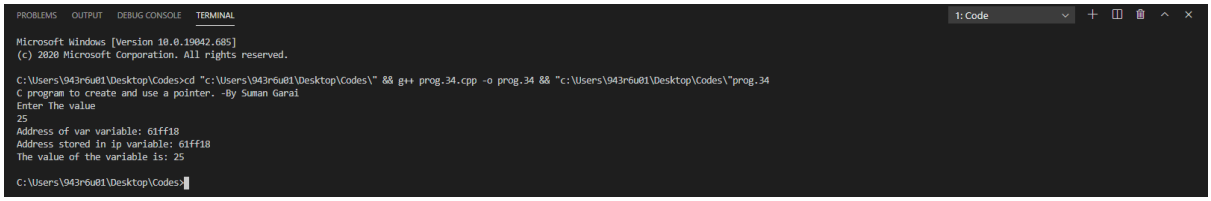
Experiment 55:

C program to create and use a pointer.

Code:

```
#include <stdio.h>
int main ()
{
    int var;
    int *ip;
    ip = &var;
    printf("C program to create and use a pointer. -
    By Suman Garai\n");
    printf("Enter The value\n");
    scanf("%d",&var);
    printf("Address of var variable: %x\n", &var );
    printf("Address stored in ip variable: %x\n", ip
    );
    printf("The value of the variable is: %d\n", *ip
    );
}
```

Output:



Experiment 56:

C program to swap the values of two variables using pointers.

Code:

```
#include <stdio.h>
void swapnum(int *i, int *j)
{
    int temp = *i;
    *i = *j;
    *j = temp;
}
int main(void)
{
    int a = 10, b = 20;
    swapnum(&a, &b);
    printf("C program to swap the values of two
    variables using pointers. -By Suman Garai\n");
    printf("a is %d and b is %d\n", a, b);
    return 0;
}
```

Output:

Experiment 57:

C program to find the area and circumference of a circle using pointers and functions.

Code:

```
#include<stdio.h>
void area_peri(float, float*, float*);
int main()
{
    float radius, area, perimeter;
    printf("C program to find the area and
    circumference of a circle using pointers and
    functions. -By Suman Garai\n");
    printf("Enter radius of Circle\n");
    scanf("%f", &radius);
    area_peri(radius, &area, &perimeter);
    printf("\nArea of Circle = %0.2f\n", area);
    printf("Perimeter of Circle = %0.2f\n",
    perimeter);
    return 0;
}
void area_peri(float r, float *a, float *p)
{
    *a = 3.14 * r * r;
    *p = 2 * 3.14 * r;
}
```

Output:

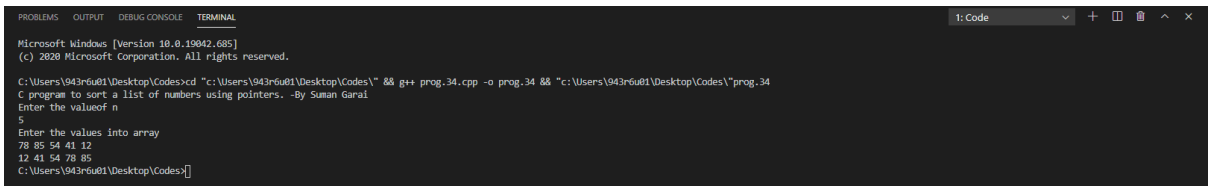
Experiment 58:

C program to sort a list of numbers using pointers.

Code:

```
#include <stdio.h>
int * sort(int n, int* ptr)
{
    int i, j, t,*p;
    p=ptr;
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (*(ptr + j) < *(ptr + i))
            {
                t = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = t;
            }
        }
    }
    return p;
}
int main()
{
    int n,arr[20],i,*ptr ;
    printf("C program to sort a list of numbers using
    pointers. -By Suman Garai\n");
    printf("Enter the valueof n\n");
    scanf("%d",&n);
    printf("Enter the values into array\n");
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    ptr=sort(n, arr);
    for (i = 0; i < n; i++)
    printf("%d ", *(ptr + i));
    return 0;
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to sort a list of numbers using pointers. -By Suman Garai
Enter the valueof n
5
Enter the values into array
78 85 54 41 12
12 41 54 78 85
C:\Users\943r6u01\Desktop\Codes>

Experiment 59:

C program to concatenate two strings using pointers and functions.

Code:

```
#include <stdio.h>
void concatenate_string(char*, char*);
int main()
{
    char original[100], add[100];
    printf("C program to concatenate two strings
    using pointers and functions. -By Suman
    Garai\n");
    printf("Enter source string: \n");
    gets(original);
    printf("Enter string to concatenate: \n");
    gets(add);
    concatenate_string(original, add);
    printf("String after concatenation: \"%s\"\n",
    original);
    return 0;
}
void concatenate_string(char *original, char *add)
{
    while(*original)
        original++;
    while(*add)
    {
        *original = *add;
        add++;
        original++;
    }
    *original = '\0';
}
```

Output:

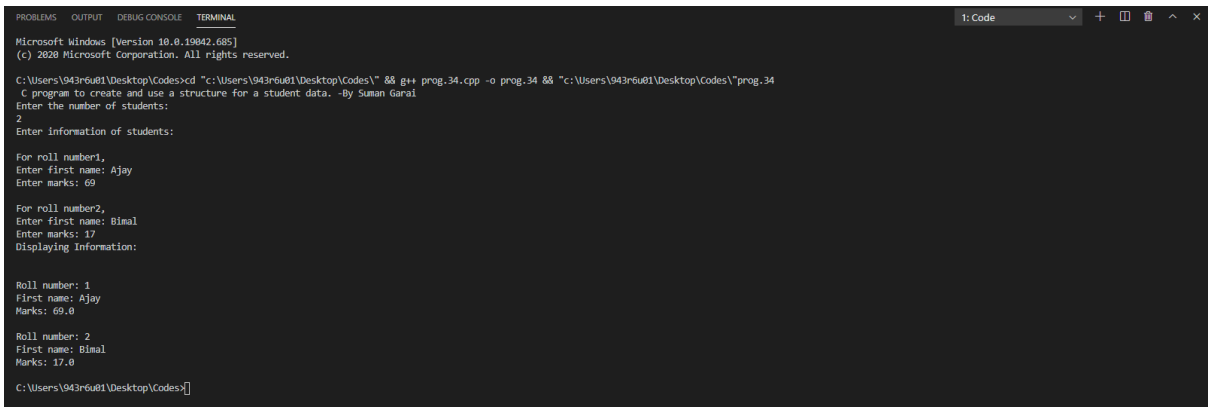
Experiment 60:

C program to create and use a structure for a student data.

Code:

```
#include <stdio.h>
struct student
{
    char firstName[50];
    int roll;
    float marks;
}   s[10];
int main()
{
    int i,n;
    printf(" C program to create and use a structure
    for a student data. -By Suman Garai\n");
    printf("Enter the number of students:\n");
    scanf("%d",&n);
    printf("Enter information of students:\n");
    for (i = 0; i < n; ++i)
    {
        s[i].roll = i + 1;
        printf("\nFor roll number%d,\n", s[i].roll);
        printf("Enter first name: ");
        scanf("%s", s[i].firstName);
        printf("Enter marks: ");
        scanf("%f", &s[i].marks);
    }
    printf("Displaying Information:\n\n");
    for (i = 0; i < n; ++i)
    {
        printf("\nRoll number: %d\n", i + 1);
        printf("First name: ");
        puts(s[i].firstName);
        printf("Marks: %.1f", s[i].marks);
        printf("\n");
    }
    return 0;
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to create and use a structure for a student data. -By Suman Garai
Enter the number of students:
2
Enter information of students:
For roll number1,
Enter first name: Ajay
Enter marks: 69
For roll number2,
Enter first name: Bimal
Enter marks: 17
Displaying Information:
Roll number: 1
First name: Ajay
Marks: 69.0
Roll number: 2
First name: Bimal
Marks: 17.0
C:\Users\943r6u01\Desktop\Codes>

Experiment 61:

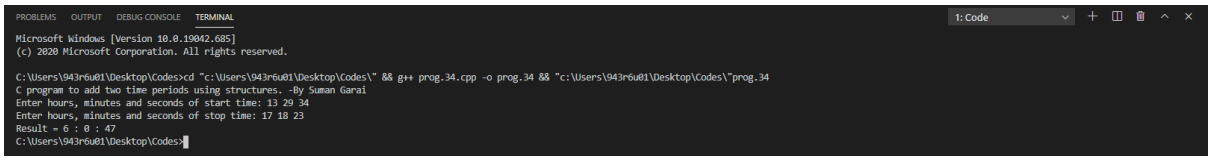
C program to add two time periods using structures.

Code:

```
#include<stdio.h>
struct time
{
    int hours;
    int minutes;
    int seconds;
};
int main()
{
    struct time start, stop, add;
    printf("C program to add two time periods using
    structures. -By Suman Garai\n");
    printf("Enter hours, minutes and seconds of start
    time: ");
    scanf("%d %d %d", &start.hours,&start.minutes,
    &start.seconds);
    printf("Enter hours, minutes and seconds of stop
    time: ");
    scanf("%d %d %d", &stop.hours,&stop.minutes,
    &stop.seconds);
    add.minutes=0;
    add.seconds=0;
    add.hours=0;
    if(start.seconds+stop.seconds>=60)
    {   add.seconds=start.seconds+stop.seconds-60;
    add.minutes=1; }
    if(start.seconds+stop.seconds<60)
    {   add.seconds=start.seconds+stop.seconds; }
    if(start.minutes+stop.minutes>=60)
    {
    add.minutes=add.minutes+start.minutes+stop.minute
    s-60; add.hours=1; }
    if(start.seconds+stop.seconds<60)
```

```
    {
    add.seconds=add.minutes+start.minutes+stop.minute
    s; }
    if(start.hours+stop.hours>=24)
    {   add.hours=add.hours+start.hours+stop.hours-
    24; }
    if(start.hours+stop.hours < 24)
    {   add.hours=add.hours+start.hours+stop.hours; }
    printf("Result = %d : %d : %d", add.hours,
    add.minutes, add.seconds);  return 0;
}
```

Output:



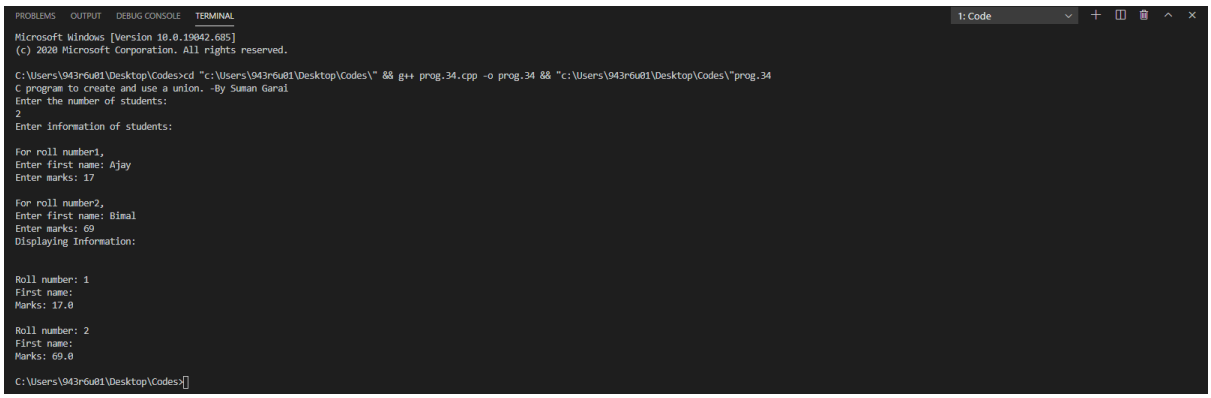
Experiment 62:

C program to find the frequency of a character in a string.

Code:

```
#include <stdio.h>
union student
{
    char firstName[50];
    int roll;
    float marks;
}   u[10];
int main()
{
    int i,n;
    printf("C program to create and use a union. -By
    Suman Garai\n");
    printf("Enter the number of students:\n");
    scanf("%d",&n);
    printf("Enter information of students:\n");
    for (i = 0; i < n; ++i)
    {
        u[i].roll = i + 1;
        printf("\nFor roll number%d,\n", u[i].roll);
        printf("Enter first name: ");
        scanf("%s", u[i].firstName);
        printf("Enter marks: ");
        scanf("%f", &u[i].marks);
    }
    printf("\nDisplaying Information:\n");
    for (i = 0; i < n; ++i)
    {
        printf("\nRoll number: %d\n", i + 1);
        printf("First name: ");
        puts(u[i].firstName);
        printf("Marks: %.1f", u[i].marks);
        printf("\n");
    }
    return 0;
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to create and use a union. -By Suman Garai
Enter the number of students:
2
Enter information of students:
For roll number1,
Enter first name: Ajay
Enter marks: 17
For roll number2,
Enter first name: Bimal
Enter marks: 69
Displaying Information:
Roll number: 1
First name:
Marks: 17.0
Roll number: 2
First name:
Marks: 69.0
C:\Users\943r6u01\Desktop\Codes>

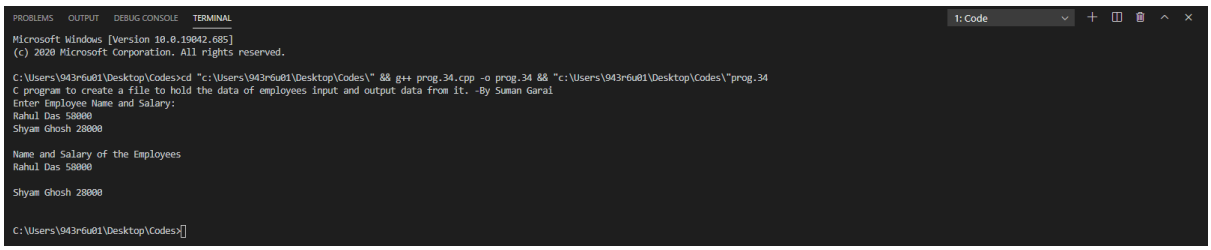
Experiment 63:

C program to create a file to hold the data of employees input and output data from it.

Code:

```
#include<stdio.h>
#include<string.h>
main( )
{
    FILE *fp;
    char s[50];
    fp=fopen("text.txt","w");
    if(fp==NULL)
    {
        puts("file opening error"); return 0;
    }
    printf("C program to create a file to hold the
    data of employees input and output data from it.
    -By Suman Garai\n");
    printf("Enter Employee Name and Salary: \n");
    while(strlen(gets(s))>0)
    {
        fputs(s,fp);
        fputs("\n",fp);
    }
    fclose(fp);
    fp=fopen("text.txt","r");
    if(fp==NULL)
    {
        puts("file opening error"); return 0;
    }
    printf("Name and Salary of the Employees\n");
    while(fgets(s,49,fp)!= NULL)
        puts(s);
    fclose(fp);
}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.34.cpp -o prog.34 && "c:\Users\943r6u01\Desktop\Codes\"prog.34
C program to create a file to hold the data of employees input and output data from it. -By Suman Garai
Enter Employee Name and Salary:
Rahul Das 58000
Shyam Ghosh 28000
Name and Salary of the Employees
Rahul Das 58000
Shyam Ghosh 28000
C:\Users\943r6u01\Desktop\Codes>

Experiment 64:

C program to write a sentence in a file and convert all lower-case alphabets to uppercase and vice versa.

Code:

```
#include<stdio.h>
#include<string.h>
#include<iostream>
main( )
{
    FILE *fp;
    char s[50],c;
    int i;
    fp=fopen("text.txt","w");
    if(fp==NULL)
    {
        puts("file opening error"); return 0;
    }
    printf("C program to write a sentence in a file
    and convert all lower case alphabets to uppercase
    and  vice versa. -By Suman Garai\n");
    printf("Enter the string\n");
    while(strlen(gets(s))>0)
    {
        fputs(s,fp);
        fputs("\n",fp);
    }
    fclose(fp);
    fp=fopen("text.txt","r");
    if(fp==NULL)
    {
        puts("file opening error"); return 0;
    }
    printf("The Output is\n");
    while(fgets(s,49,fp)!= NULL)
    {
        i=0;
        while(s[i]!='\0')
```

```
        {
            c=s[i];
            if(isupper(c))
                printf("%c",tolower(c));
            else if(islower(c))
                printf("%c",toupper(c));
            else
                printf("%c",c);
            i++;
        }
    }
    fclose(fp);
}
```

Output:

-- THE END --