SUBBALAKSHMI LAKSHMIPATHY COLLEGE OF SCIENCE

Affiliated to Madurai Kamaraj University

Re-accredited with 'A+' Status by NAAC

TVR NAGAR, ARUPPUKOTTAI ROAD

MADURAI-625022

DEPARTMENT OF COMPUTER SCIENCE (SSS & DSA)

Programming in C-Lab – 24DCS106P

DEPARTMENT OF COMPUTER SCIENCE

(SSS&DSA)

RECORD NOTE BOOK

2024 - 2025

SUBBALAKSHMI LAKSHMIPATHY COLLEGE OF SCIENCE

"Affiliated to Madurai Kamaraj University and Re-accredited with 'A+' Status by NAAC"

TVR NAGAR, ARUPPUKOTTAI ROAD, MADURAI-625022

Name :

Roll no :

DEPARTMENT OF COMPUTER SCIENCE (SSS & DSA)

CERTIFICATE

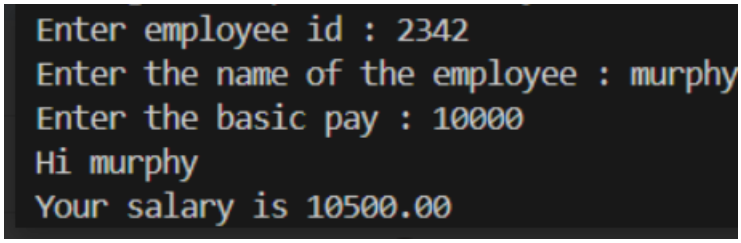
This is to certify that _____ of I B.Sc. Computer Science (DSA) has successfully completed her Practical's in PROGRAMMING IN C LAB from the college during the year 2024-2025 and submitted for practical examinations held on_____.

Faculty In-charge **HOD**

Internal Examiner **External Examiner**

INDEX

OUTPUT:


Enter employee id : 2342
Enter the name of the employee : murphy
Enter the basic pay : 10000
Hi murphy
Your salary is 10500.00

S.NO:1	CALCULATE SALARY OF AN EMPLOYEE
17/07/24	

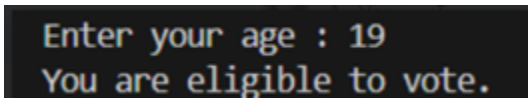
AIM:

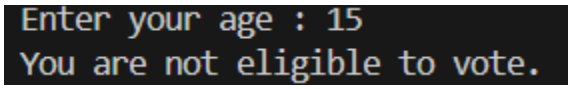
Write a C program to calculate salary of an employee .

SOURCE OF CODE:

```
//to find the netpay of a worker
#include<stdio.h>
void main()
{
   int empid,bp;
   float hra,da,ta,it,pf,netpay;
   char name[40];

   printf("Enter employee id : ");
   scanf("%d",&empid);
   printf("Enter the name of the employee : ");
   scanf("%s",name);
   printf("Enter the basic pay : ");
   scanf("%d",&bp);
   //calculations
   hra = 10 / 100.0 * bp;
   da = 8 / 100.0 * bp;
   ta = 9 / 100.0 * bp;
   it = 10 / 100.0 * bp;
   pf = 12 / 100.0 * bp;

   netpay = bp + hra + da + ta - (it + pf);

   //output
   printf("Hi %s\n",name);
   printf("Your salary is %.2f",netpay);

}
```

OUTPUT:


Enter your age : 19
You are eligible to vote.


Enter your age : 15
You are not eligible to vote.

S.NO:2	VERIFICATION OF AGE FOR VOTING
20/07/24	

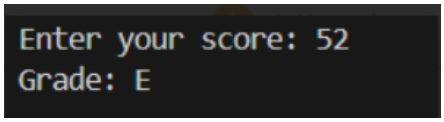
AIM:

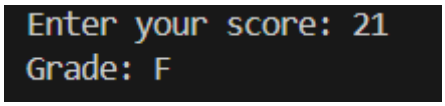
Write a C program to check the eligibility of voting

SOURCE OF CODE:

```
//to check vote eligibility
#include<stdio.h>
void main()
{
   int age;
   printf("Enter your age : ");
   scanf("%d",&age);

   if (age >= 18)
      printf("You are eligible to vote. ");
   else
      printf("You are not eligible to vote.");
}
```

OUTPUT:





S.NO:3	GRADE CALCULATION
22/07/24	

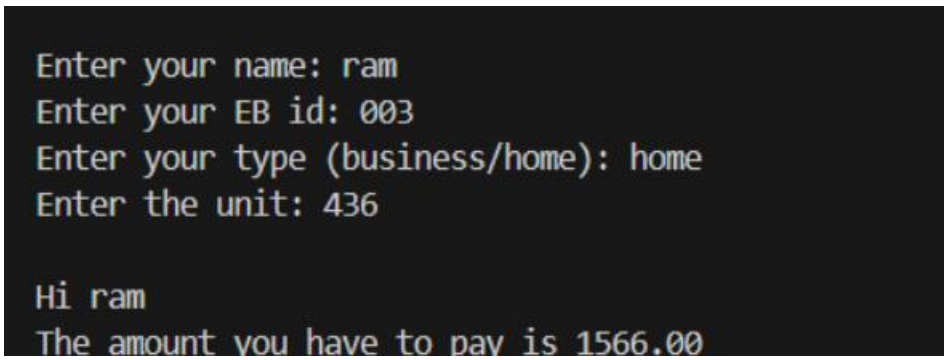
AIM:

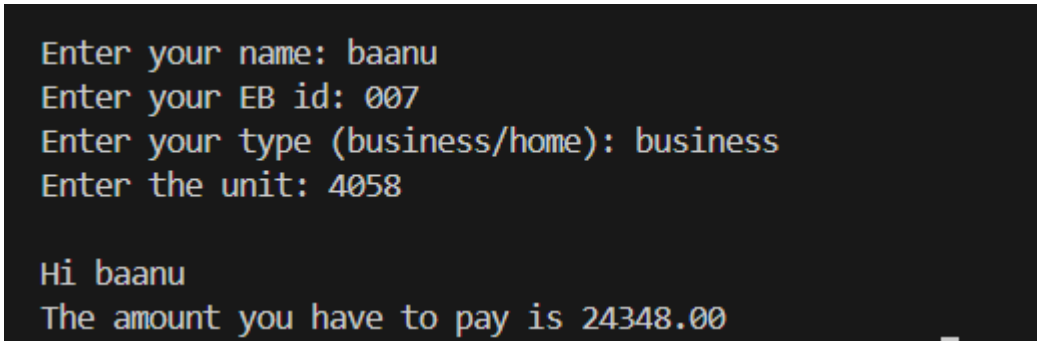
Write a C program to calculate grade of the students.

SOURCE OF CODE:

```
//to find the grade of the student
#include <stdio.h>

int main()
{
   int score;

   //accept the score from the user
   a:printf("Enter your score: ");
   scanf("%d", &score);

   if (score > 100)
   {
      printf("Invalid score. Please enter a score between 0 and 100.\n");
      goto a;
   }
   else if (score >= 90)
      printf("Grade: A\n");
   else if (score >= 80)
      printf("Grade: B\n");
   else if (score >= 70)
      printf("Grade: C\n");
   else if (score >= 60)
      printf("Grade: D\n");
   else if (score >= 50)
      printf("Grade: E\n");
   else if (score >= 0)
      printf("Grade: F\n");

   return 0;
}
```

OUTPUT:


Enter your name: ram
Enter your EB id: 003
Enter your type (business/home): home
Enter the unit: 436
Hi ram
The amount you have to pay is 1566.00


Enter your name: baanu
Enter your EB id: 007
Enter your type (business/home): business
Enter the unit: 4058
Hi baanu
The amount you have to pay is 24348.00

S.NO:4	EB-BILL CALCULATION
23/07/24	

AIM:

To write a C program to find the eb-bill amount.

SOURCE OF CODE:

```
//to find the EB-bill amount for the given unit
#include<stdio.h>
#include<string.h>
int main() {
    char name[30], type[9];
    int id, unit;
    float amt = 0;

   //taking input from the user
    printf("Enter your name: ");
    scanf("%s", name);
    printf("Enter your EB id: ");
    scanf("%d", &id);
    printf("Enter your type (business/home): ");
    scanf("%s", type);
    printf("Enter the unit: ");
    scanf("%d", &unit);

   //calculations
    if (strcmp(type, "business") == 0) {
        if (unit >= 1000)
            amt = 6 * unit;
        else if (unit >= 500)
            amt = 4 * unit;
        else
            amt = 2 * unit;
    }else if (strcmp(type, "home") == 0)
     {
        if (unit > 1000)
            amt = (unit - 1000) * 11.00 + 200 * 10.00 + 200 * 9.00 + 100 * 8.00 + 100 *    6.00 +
300 * 4.50;
        else if (unit > 800)
            amt = (unit - 800) * 10.00 + 200 * 9.00 + 100 * 8.00 + 100 * 6.00 + 300 * 4.50;
        else if (unit > 600)
            amt = (unit - 600) * 9.00 + 100 * 8.00 + 100 * 6.00 + 300 * 4.50;
        else if (unit > 500)
            amt = (unit - 500) * 8.00 + 100 * 6.00 + 300 * 4.50;
        else if (unit > 400)
            amt = (unit - 400) * 6.00 + 300 * 4.50;
        else if (unit > 100)
```

```
            amt = (unit - 100) * 4.50;
    }

//output
    printf("\nHi %s", name);
    printf("\nThe amount you have to pay is %.2f", amt);
    return 0;
}
```

OUTPUT:

```
Enter a number between 1 and 12: 23

Invalid input
Enter a number between 1 and 12: 12
December
```

S.NO:5	DISPLAYING EQUIVALENT MONTH
29/07/24	

AIM:

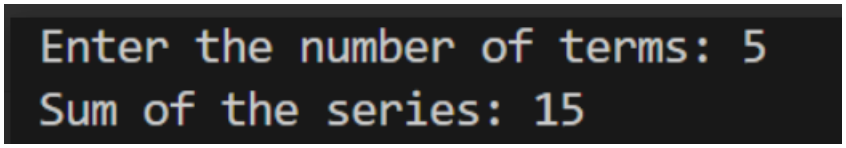
Write a C program to display equivalent months.

SOURCE OF CODE:

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

   a : printf("Enter a number between 1 and 12: ");
   scanf("%d", &month);

   switch(month){
      case 1:
         printf("January");
         break;
      case 2:
         printf("February");
         break;
      case 3:
         printf("March");
         break;
      case 4:
         printf("April");
         break;
      case 5:
         printf("May");
         break;
      case 6:
         printf("June");
         break;
      case 7:
         printf("July");
         break;
      case 8:
         printf("August");
```

```
    break;
      case 9:
         printf("September");
         break;
      case 10:
         printf("October");
         break;
      case 11:
         printf("November");
         break;
      case 12:
         printf("December");
         break;
      default:
         printf("\nInvalid input\n");
         goto a;
         break;
   }

   return 0;
}
```

OUTPUT:



S.NO:6	SUM OF SERIES
08/08/24	

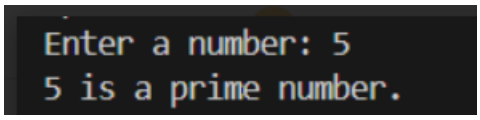
AIM:

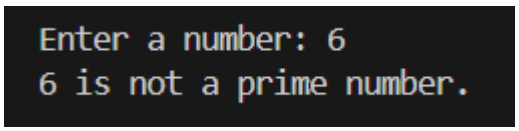
Write a C program to find sum of series.

SOURCE OF CODE:

```
//to find the sum of the series
#include <stdio.h>
int main()
{
   int n, sum = 0, i = 1;

   printf("Enter the number of terms: ");
   scanf("%d", &n);

   while(i <= n)
   {
      sum += i;
      i++;
   }

   printf("Sum of the series: %d\n", sum);

   return 0;
}
```

OUTPUT:


Enter a number: 5
5 is a prime number.


Enter a number: 6
6 is not a prime number.

S.NO:7	PRIME NUMBERS
12/08/24	

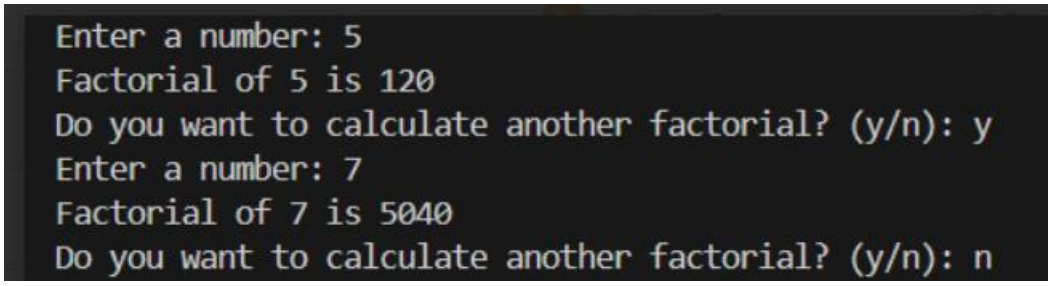
AIM:

Write a C program to find prime numbers.

SOURCE OF CODE:

```
#include <stdio.h>

int main()
{
   int n, i, isPrime = 1;

   printf("Enter a number: ");
   scanf("%d", &n);

   if (n <= 1)
   {
      isPrime = 0;
   }
   else
   {
      for (i = 2; i <= n / 2; i++)
      {
         if (n % i == 0)
         {
            isPrime = 0;
            break;
         }
      }
   }

   if (isPrime)
      printf("%d is a prime number.\n", n);
   else
      printf("%d is not a prime number.\n", n);

   return 0;
}
```

OUTPUT:


Enter a number: 5
Factorial of 5 is 120
Do you want to calculate another factorial? (y/n): y
Enter a number: 7
Factorial of 7 is 5040
Do you want to calculate another factorial? (y/n): n

S.NO:8	FINDING FACTORIAL IN A LOOP
14/08/24	

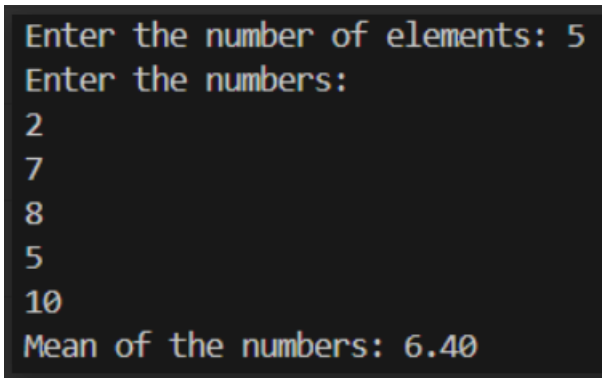
AIM:

Write a C program to find a factorial of the given number.

SOURCE OF CODE:

```
//to find the factorial to the given number until the user says no
#include <stdio.h>

int main()
{
   int n, i;
   char choice;

   do
   {
      int factorial = 1; // Initialize factorial variable for each iteration

      printf("Enter a number: ");
      scanf("%d", &n);

      for(i = 1; i <= n; i++)
      {
         factorial *= i;
      }

      printf("Factorial of %d is %d\n", n, factorial);

      printf("Do you want to calculate another factorial? (y/n): ");
      scanf(" %c", &choice);
  // Read the user's choice (note the space before %c to consume any newline)

   } while(choice == 'y' || choice == 'Y');

   return 0;
}
```

OUTPUT:


Enter the number of elements: 5
Enter the numbers:
2
7
8
5
10
Mean of the numbers: 6.40

S.NO:9	FINDING MEAN OF NUMBERS
16/08/24	

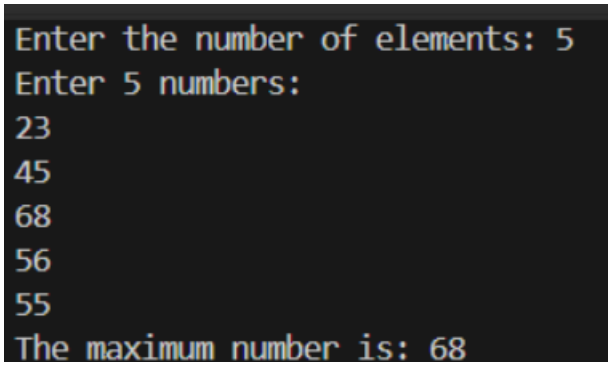
AIM:

Write a C program to find a mean of a given number .

SOURCE CODE:

```
//to find the minimum of given n numbers
#include <stdio.h>

int main()
{
   int n, i;
   float sum = 0, mean, arr[100];

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   printf("Enter the numbers:\n");
   for(i = 0; i < n; i++)
   {
      scanf("%f", &arr[i]);
   }
   for(i = 0; i < n; i++)
   {
      sum += arr[i];
   }

   mean = sum / n;

   printf("Mean of the numbers: %.2f\n", mean);

   return 0;
}
```

OUTPUT:


Enter the number of elements: 5
Enter 5 numbers:
23
45
68
56
55
The maximum number is: 68

S.NO:10	MAXIMUM OF NUMBERS
26/08/24	

AIM:

Write a C program to find the maximum of a given number.

SOURCE CODE:

```
//to find the maximum of the given numbers
#include <stdio.h>

int main()
{
   int n, i, max, arr[100];

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   printf("Enter %d numbers:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }

   max = arr[0];

   for(i = 1; i < n; i++)
   {
      if(arr[i] > max)
      {
         max = arr[i];
      }
   }

   printf("The maximum number is: %d\n", max);

   return 0;
}
```

OUTPUT:

```
Enter the number of elements: 5
Enter 5 numbers:
234
4
32
4
1
The minimum number is: 1
```

S.NO:11	MINIMUM OF NUMBERS
05/09/24	

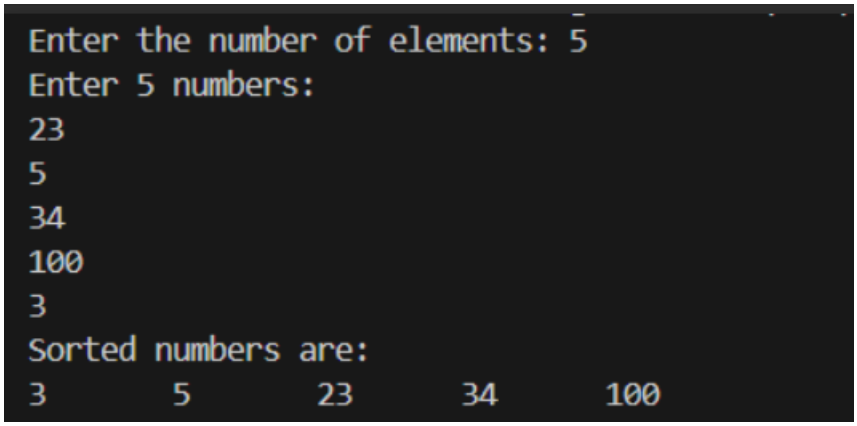
AIM:

Write a C program to find the minimum of a given number.

SOURCE CODE:

```
//to find the minimum of given n numbers
#include <stdio.h>

int main()
{
   int n, i, min;
   int arr[100];

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   printf("Enter %d numbers:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }

   min = arr[0];

   for(i = 1; i < n; i++)
   {
      if(arr[i] < min)
      {
         min = arr[i];
      }
   }

   printf("The minimum number is: %d\n", min);

   return 0;
}
```

OUTPUT:


Enter the number of elements: 5
Enter 5 numbers:
23
5
34
100
3
Sorted numbers are:
3 5 23 34 100

S.NO:12	SORTING NUMBERS
05/09/24	

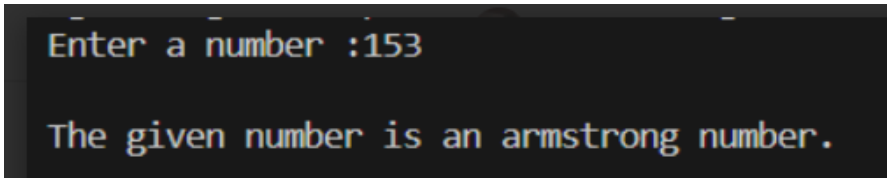
AIM:

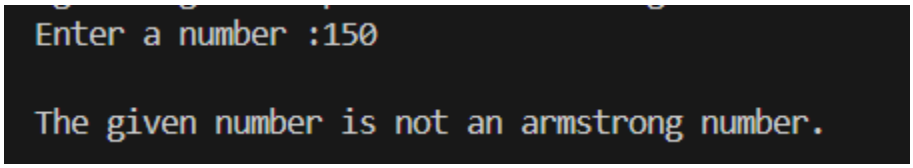
Write a C program to sort the given n numbers.

SOURCE OF CODE:

```
//sorting
#include <stdio.h>
int main()
{
   int n, i, j, temp;
   int arr[100];

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   printf("Enter %d numbers:\n", n);
   for(i = 0; i < n; i++){
      scanf("%d", &arr[i]);
   }

   for(i = 0; i < n-1; i++)
   {
      for(j = i+1; j < n; j++)
      {
         if(arr[i] > arr[j]){
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
         }
      }
   }

   printf("Sorted numbers are:\n");
   for(i = 0; i < n; i++)
   {
      printf("%d\t", arr[i]);
   }
   return 0;
}
```

OUTPUT:


Enter a number :153
The given number is an armstrong number.


Enter a number :150
The given number is not an armstrong number.

S.NO:13	ARMSTRONG NUMBER
06/09/24	

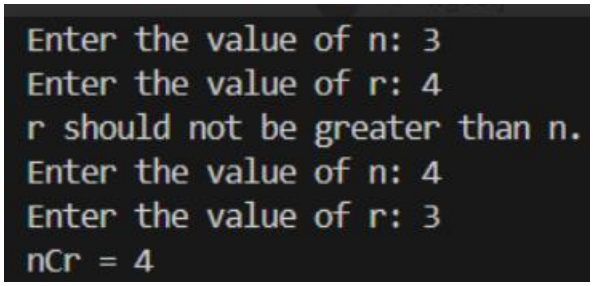
AIM:

Write a C program to find the Armstrong number.

SOURCE OF CODE:

```
//to check the given numbers are Armstrong or not
#include<stdio.h>
#include<math.h>
int count(int no){
   int n,count = 0;
   //to count the numbers
   while (no != 0){
      no /= 10;
      count++;
   }
   return count;
}
int armstrong(int no){
   int i,rem,ans=0;
   int c = count(no);
   //to find the armstrong number
   for (i = 0; i < c; i++){
      rem = no % 10;
      ans = ans + pow(rem, c);
      no /= 10;
   }
   return ans;
}
int main(){
   int no;
   printf ("Enter a number :");
   scanf("%d",&no);
   if (armstrong(no) == no)
     printf("\nThe given number is an armstrong number.\n");
   else
     printf("\nThe given number is not an armstrong number.\n");
   return 0;
}
```

OUTPUT:


Enter the value of n: 3
Enter the value of r: 4
r should not be greater than n.
Enter the value of n: 4
Enter the value of r: 3
nCr = 4

S.NO:14	FINDING NCR
09/09/24	

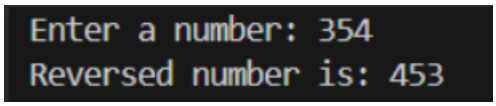
AIM:

Write a C program to find the NCR value for the given values.

SOURCE OF CODE:

```
//NCR calculation
#include <stdio.h>
int factorial(int n)
{
   int i,fact = 1;
   for (i = 1; i <= n; i++)
   {
      fact *= i;
   }
   return fact;
}

int main()
{
   int n, r;
   int nCr;

   a:printf("Enter the value of n: ");
   scanf("%d", &n);
   printf("Enter the value of r: ");
   scanf("%d", &r);

   if (r > n)
   {
      printf("r should not be greater than n.\n");
      goto a;
   }

   nCr = factorial(n) / (factorial(r) * factorial(n - r));

   printf("nCr = %d\n", nCr);

   return 0;
}
```

OUTPUT:



S.NO:15	REVERSE OF THE GIVEN NUMBER
10/09/24	

AIM:

Write a C program to find the reverse of the given numbers.

SOURCE OF CODE:

```
// to find the reverse of the given number
#include <stdio.h>

int reverse(int num)
{
   int rev = 0;
   while (num > 0)
   {
      rev = rev * 10 + num % 10;
      num /= 10;
   }
   return rev;
}

int main()
{
   int num;

   printf("Enter a number: ");
   scanf("%d", &num);

   printf("Reversed number is: %d\n", reverse(num));

   return 0;
}
```

OUTPUT:

```
Enter a number: 100
100 is not an Adam number
```

```
Enter a number: 101
101 is an Adam number
```

S.NO:16	ADAM NUMBER
10/09/24	

AIM:

Write a C program to find the Adam number.

SOURCE OF CODE:

```
//to find the given number is Adam number or not
 #include <stdio.h>
int reverse(int num) //a function only to reverse the given number
{
   int rev = 0;
   while (num > 0){
      rev = rev * 10 + num % 10;
      num /= 10;
   }
   return rev;
}
int isAdamNumber(int num) {
   int rev, sq1, sq2, revsqr;
   rev = reverse(num);
   sq1 = num * num;
   sq2 = rev * rev;
   revsqr = reverse(sq2);
   if (sq1 == revsqr)
      return 0;
   else
      return 1;
}

int main() {
   int num;
   printf("Enter a number: ");
   scanf("%d", &num);
   if (0 == isAdamNumber(num))
      printf("%d is an Adam number\n", num);
   else
      printf("%d is not an Adam number\n", num);

   return 0;
}
```

OUTPUT:

```
Enter a number :6
6 is a perfect number.
```

```
Enter a number :7
7 is not a perfect number.
```

S.NO:17	PERFECT NUMBER
12/09/24	

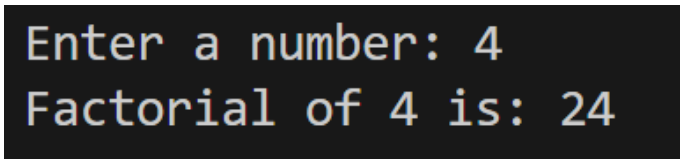
AIM:

Write a C program to find the perfect number.

SOURCE OF CODE:

```
//to find the given number is a perfect number or not
#include<stdio.h>
int isperfect (int no)
{
   int i,check=0;
   for (i = 1; i < no; i++)
   {
      if (no % i == 0)
        check += i;
   }
   return check;
}
int main()
{
   int no;

   printf("Enter a number :");
   scanf("%d",&no);

   if (isperfect(no) == no)
     printf("%d is a perfect number.",no);
   else
     printf("%d is not a perfect number.",no);
   return 0;
}
```

OUTPUT:



S.NO:18	FACTORIAL
13/09/24	

AIM:

Write a C program to find the factorial of the given number.

SOURCE OF CODE:

```
//to find the factorial using function
#include <stdio.h>

long factorial(int num)
{
   long i, result = 1;
   for (i = 1; i <= num; i++)
   {
      result *= i;
   }
   return result;
}

int main()
{
   int num;

   printf("Enter a number: ");
   scanf("%d", &num);

   if (num < 0)
      printf("Factorial is not defined for negative numbers.\n");
   else
      printf("Factorial of %d is: %ld\n", num, factorial(num));

   return 0;
}
```

OUTPUT:

```
Enter the number of rows and columns of the matrix :
2
2
Enter the elements of the first matrix :
2
5
7
5
Enter the elements of the second matrix:
3
4
1
0
The first matrix is :
2 5
7 5
The second matrix is :
3 4
1 0
Sum of the matrices:
5 9
8 5
```

S.NO:19	MATRIX ADDITION
14/09/24	

AIM:

Write a C program to find the sum of two matrix.

SOURCE OF CODE:

```
//matrix addition
#include <stdio.h>

int main()
{
   int rows, cols, i, j;
   int matrix1[100][100], matrix2[100][100], sum[100][100];
   // Input dimensions of the matrix
   printf("Enter the number of rows and columns of the matrix : \n");
   scanf("%d %d", &rows, &cols);

   // Input elements of the first matrix
   printf("Enter the elements of the first matrix :\n");
   for (i = 0; i < rows; i++)
   {
      for (j = 0; j < cols; j++)
      {
         scanf("%d", &matrix1[i][j]);
      }
   }

   // Input elements of the second matrix
   printf("Enter the elements of the second matrix:\n");
   for (i = 0; i < rows; i++)
   {
      for (j = 0; j < cols; j++)
      {
         scanf("%d", &matrix2[i][j]);
      }
   }

   // Adding the two matrices
   for (i = 0; i < rows; i++)
   {
```

```
 for (j = 0; j < cols; j++)
        {
            sum[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
// Showing the inputted matrices
    printf("The first matrix is : \n");
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", matrix1[i][j]);
        }
        printf("\n");
    }
    printf("The second matrix is : \n");
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", matrix2[i][j]);
        }
        printf("\n");
    }

    // Display the result
    printf("Sum of the matrices:\n");
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT:

```
Enter rows and columns of first matrix :
2 3
Enter rows and columns of second matrix :
3 2
Enter elements of first matrix:
1
2
3
4
5
6
Enter elements of second matrix:
7
8
9
10
11
12
The first matrix is :
1 2 3
4 5 6
The second matrix is :
7 8
9 10
11 12
The resultant matrix :
58 64
139 154
```

S.NO:20	MATRIX MULTIPLICATION
26/09/24	

AIM:

Write a C program to find the multiplication two matrix.

SOURCE OF CODE:

```
//matrix multiplication
#include<stdio.h>

void main()
{
   int r1,c1,r2,c2,i,j,k;

   int matrix1[100][100], matrix2[100][100], ans[100][100];

   // Input dimensions of matrix
   a:printf("Enter rows and columns of first matrix : \n");
   scanf("%d %d",&r1,&c1);

   printf("Enter rows and columns of second matrix : \n");
   scanf("%d %d",&r2,&c2);

   // Verifying whether the conditions for matrix multiplication are satisfied.
   if (c1 != r2)
   {
      printf("Matrix multiplication is not possible.\n");
      goto a;
   }

   // Input matrix 1
   printf("Enter elements of first matrix:\n");
   for (i = 0; i < r1; ++i)
   {
      for (j = 0; j < c1; ++j)
      {
         scanf("%d", &matrix1[i][j]);
      }
   }
```

```
 // Input matrix 2
 printf("Enter elements of second matrix:\n");
 for (i = 0; i < r2; ++i)
 {
    for (j = 0; j < c2; ++j)
    {
       scanf("%d", &matrix2[i][j]);
    }
 }
// Displaying matrices.
 printf("The first matrix is : \n");
 for (i = 0; i < r1; i++)
  {
    for (j = 0; j < c1; j++)
    {
       printf("%d ", matrix1[i][j]);
    }
    printf("\n");
 }
 printf("The second matrix is : \n");
 for (i = 0; i < r2; i++)
 {
    for (j = 0; j < c2; j++)
    {
       printf("%d ", matrix2[i][j]);
    }
    printf("\n");
 }

 // Multiplying the matrices.
 for (i = 0; i < r1; i++)
 {
    for (j = 0; j < c2; j++)
    {
       ans[i][j] = 0;
       for (k = 0; k < c1; k++)
       {
          ans[i][j] += matrix1[i][k] * matrix2[k][j];
       }
    }
 }
```

```
    // Display the resultant matrix.
    printf("The resultant matrix : \n");
     for (i = 0; i < r1; i++)
     {
        for (j = 0; j < c2; j++)
        {
            printf("%d ", ans[i][j]);
        }
        printf("\n");
    }

}
```

OUTPUT:

```
Etner rows and columns of the matrix :
3 3
Enter the elements of the matrix :
1
2
3
4
5
6
7
8
9

The given matrix is
1       2       3
4       5       6
7       8       9

The transporse of the given matrix is
1       4       7
2       5       8
3       6       9
```

S.NO:21	TRANSPOSE OF MATRIX
27/09/24	

AIM:

Write a C program to transpose the given matrix.

SOURCE OF CODE:

```
//matrix transpose
#include<stdio.h>
int main()
{
   int matrix[100][100],row,col,i,j;

   printf("Etner rows and columns of the matrix : \n");
   scanf("%d %d",&row,&col);

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

   printf("\nThe given matrix is \n");
   for (i = 0; i < row; i++){
      for (j = 0; j < col; j++){
         printf("%d\t",matrix[i][j]);
      }
      printf("\n");
   }
   printf("\nThe transpose of the given matrix is \n");
   for (i = 0; i < row; i++){
      for (j = 0; j< col; j++){
         printf("%d\t",matrix[j][i]);
      }
      printf("\n");
   }
   return 0;
}
```

OUTPUT:

```
Enter the number of names: 4
Enter 4 names:
Prakash
Sam
Priya
Bharathi


Sorted names are:

Bharathi
Prakash
Priya
Sam
```

S.NO:22	NAMES SORTING
30/09/24	

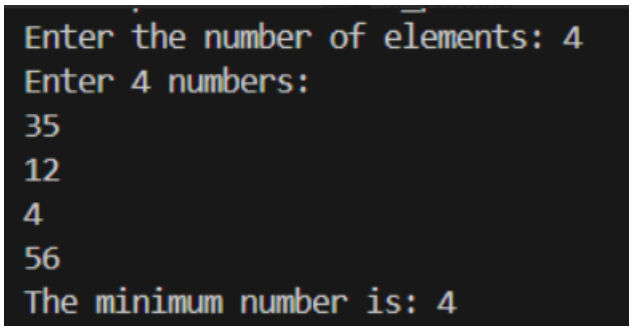
AIM:

Write a c program to sort the given names.

SOURCE OF CODE:

```
//sort the given names
#include <stdio.h>
#include <string.h>
int main() {
   int n,i;
   char names[60][50],temp[50];

   printf("Enter the number of names: ");
   scanf("%d", &n);

   printf("Enter %d names:\n", n);
   for (i = 0; i < n; i++){
      scanf("%s",names[i]);
   }

   // Bubble sort
   for (int i = 0; i < n ; i++) {
      for (int j = i + 1; j < n ; j++) {
         if (strcmp(names[i], names[j]) > 0) {
            strcpy(temp, names[i]);
            strcpy(names[i], names[j]);
            strcpy(names[j], temp);
         }
      }
   }

   printf("\nSorted names are:\n\n");
   for (int i = 0; i < n; i++) {
      printf("%s\n", names[i]);
   }

   return 0;
}
```

OUTPUT:


Enter the number of elements: 4
Enter 4 numbers:
35
12
4
56
The minimum number is: 4

S.NO:23	MINIMUM USING POINTERS
01/10/24	

AIM:

Write a c program to find the minimum of given n numbers using pointers.

SOURCE OF CODE:

```
//to find the minimum of given n numbers with the help of pointers
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int n, i, min;
   int *arr;

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   arr = (int *)malloc(n * sizeof(int));

   printf("Enter %d numbers:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", (arr + i));
   }

   min = *arr;

   for(i = 1; i < n; i++)
   {
      if(*(arr + i) < min)
      {
         min = *(arr + i);
      }
   }
   printf("The minimum number is: %d\n", min);
   return 0;
}
```

OUTPUT:

```
Enter the number of elements: 4
Enter 4 numbers:
3
5
34
5
The maximum number is: 34
```

S.NO:24	MAXIMUM USING POINTERS
04/10/24	

AIM:

Write a c program to find the maximum of the given numbers with pointers.

SOURCE OF CODE:

```
//to find the maximum of given n numbers
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int n, i, max;
   int *arr;

   printf("Enter the number of elements: ");
   scanf("%d", &n);

   arr = (int *)malloc(n * sizeof(int));

   printf("Enter %d numbers:\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", (arr + i));
   }

   max = *arr;

   for(i = 1; i < n; i++)
   {
      if(*(arr + i) > max)
      {
         max = *(arr + i);
      }
   }

   printf("The maximum number is: %d\n", max);

   return 0;
}
```

OUTPUT:

```
Enter the number of elements: 4
Enter 4 numbers:
2
53
3
47

Sorted numbers:
2 3 47 53
```

S.NO:25	SORTING USING POINTERS
08/10/24	

AIM:

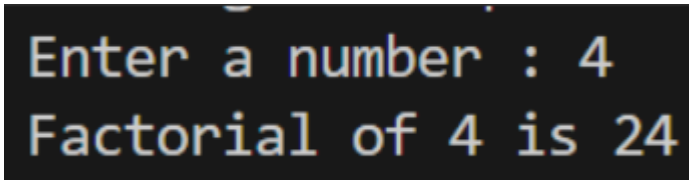
Write a c program to sort the given numbers using pointers.

SOURCE OF CODE:

```
//sorting numbers with pointers
#include <stdio.h>
#include <stdlib.h>

int main() {
   int n, *arr,temp,i,j;
   printf("Enter the number of elements: ");
   scanf("%d", &n);


   arr = (int *)malloc(n * sizeof(int));

   if (arr == NULL) {
      printf("Memory allocation failed!\n");
      return 1;
   }

   printf("Enter %d numbers: \n", n);
   for (int i = 0; i < n; i++) {
      scanf("%d", (arr+i));
   }

   for (int i = 0; i < n - 1; i++) {
      for (int j = i + 1; j < n; j++) {
         if (*(arr + j) < *(arr + i))
         {

            temp = *(arr + i);
            *(arr + i) = *(arr + j);
            *(arr + j) = temp;
         }
      }
   }
```

```
    //display output
    printf("\nSorted numbers: \n");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", *(arr + i));
    }
    return 0;
}
```

OUTPUT:



S.NO:26	FACTORIAL USING RECURSION
15/10/24	

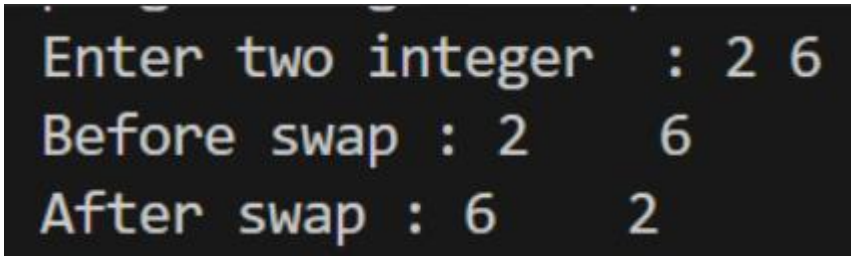
AIM:

Write a c program to find factorial using recursion.

SOURCE OF CODE:

```
//to find factorial using recursion
#include<stdio.h>

long fact(int n)
{
   if(n == 0 || n == 1)
      return 1;
   else
      return n * fact(n - 1);
}
int main()
{
   int num;
   printf("Enter a number : ");
   scanf("%d",&num);

   if(num < 0)
      printf("Not possible. ");
   else
      printf("Factorial of %d is %d",num,fact(num));

   return 0;
}
```

OUTPUT:



S.NO:27	SWAP BY CALL BY REFERENCE
16/10/24	

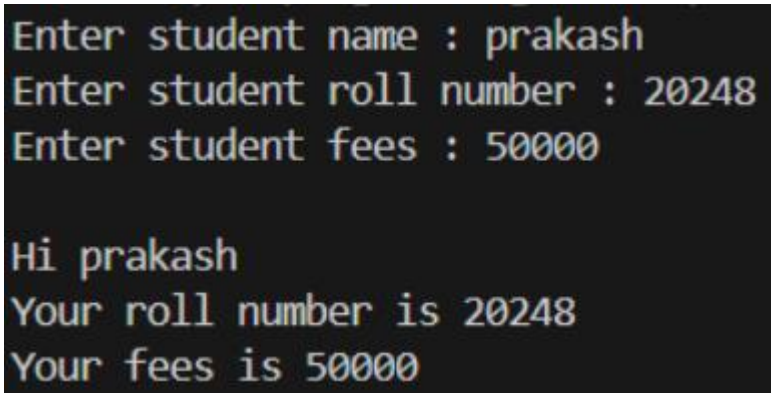
AIM:

Write a C program to swap by using call by reference method.

SOURCE OF CODE:

```
#include<stdio.h>
//function to swap two integer using pointers
void swap(int *n1,int *n2)
{
   int temp;
   temp = *n1;
   *n1 = *n2;
   *n2 = temp;

}
int main()
{
   int n1,n2;
    //asking the user to input two integers
   printf("Enter two integer  : ");
   scanf("%d %d",&n1,&n2);
    //display the values before swapping
   printf("Before swap : %d    %d\n",n1,n2);
    //call the swap function
   swap(&n1,&n2);
    //display the values after swapping
   printf("After swap : %d    %d",n1,n2);

   return 0;
}
```

OUTPUT:


Enter student name : prakash
Enter student roll number : 20248
Enter student fees : 50000
Hi prakash
Your roll number is 20248
Your fees is 50000

S.NO:28	STUDENTS INFORMATION USING STRUCTURE
10/10/24	

AIM:

Write a C program to accept and display student details using structure.

SOURCE OF CODE:

```
#include<stdio.h>
// Defining a structure 'student' to store student details
struct student
{
   int rno;
   char name[20];
   long int fees;

}student;
int main()
{
    // Declaring a structure variable 's' of type 'student'
   struct student s;

    // Prompting user to enter student details
   printf("Enter student name : ");
   scanf("%s",s.name);

   printf("Enter student roll number : ");
   scanf("%d",&s.rno);

   printf("Enter student fees : ");
   scanf("%li",&s.fees);

    // Displaying the output
   printf("\nHi %s\n",s.name);
   printf("Your roll number is %d\n",s.rno);
   printf("Your fees is %li",s.fees);

   return 0;
}
```

OUTPUT:

```
1. Store the student data
2. Retrieve the student data
0. Exit
Enter your choice: 1
Enter roll no  :  202
Enter name     :  prakash
Enter 3 subject marks   :  100 99 87
1. Store the student data
2. Retrieve the student data
0. Exit
Enter your choice: 2
Roll No    Name   Total     Percentage
202     prakash         286     95.33
1. Store the student data
2. Retrieve the student data
0. Exit
Enter your choice: 1
Enter roll no  :  203
Enter name     :  pavan
Enter 3 subject marks   :  77 56 44
1. Store the student data
2. Retrieve the student data
0. Exit
Enter your choice: 2
Roll No    Name   Total     Percentage
202     prakash         286     95.33
203     pavan           177     59.00
1. Store the student data
2. Retrieve the student data
0. Exit
Enter your choice: 0
```

S.NO:29	STORE AND RETRIEVE DATA FROM A FILE
19/10/24	

```
// Program to store and retrieve student data to and from a file
#include <stdio.h>
#include <stdlib.h>
//assign a global structure
typedef struct student{
  int rno;
  char sname[40];
  int m1, m2, m3;
  int total;
  float per;
} student;
//function to store data from user to a file
void store(){
  FILE *fp;
  student s1;
  fp = fopen("stud.txt", "ab");

  if (fp == NULL){
    printf("Error opening file!\n");
    return;
  }

  printf("Enter roll no  :  ");
  scanf("%d", &s1.rno);
  printf("Enter name     :  ");
  scanf("%s", s1.sname);
  printf("Enter 3 subject marks   :  ");
  scanf("%d %d %d", &s1.m1, &s1.m2, &s1.m3);
  s1.total = s1.m1 + s1.m2 + s1.m3;
  s1.per = s1.total / 3.0;

  fwrite(&s1, sizeof(student), 1, fp);
  fclose(fp);
}

//function to display the data from the file
void display(){
  FILE *fp;
  student s1;
```

```
 fp = fopen("stud.txt", "rb");
  if (fp == NULL){
    printf("Can't open the file\n");
    return;
  }

  printf("Roll No    Name   Total     Percentage   \n");

  while (fread(&s1, sizeof(student), 1, fp)){
    printf("%d  \t%s    \t%d\t%.2f\n", s1.rno, s1.sname, s1.total, s1.per);
  }

  fclose(fp);
}

int main(){
  int ch;

  do{
    printf("1. Store the student data\n");
    printf("2. Retrieve the student data\n");
    printf("0. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &ch);

    switch (ch){
    case 1:
      store();
      break;
    case 2:
      display();
      break;
    case 0:
      exit(0);
      break;
    default:
      printf("Wrong choice\n");
      break;
    }
  } while (ch != 0);

  return 0;
}
```