EX.NO:1a **DATE:**	**PROGRAM USING I/O STATEMENTS AND EXPRESSIONS**

AIM:

To find the VAT amount and the total value of the product.

ALGORITHM:

1.Start the program.
2.Read the item Details.
3.Read discount and tax % details.
4.Calculate amount qty x value.
5.Calculate discount (amount x discount)/100.
6.Calculate taxamt (subtot x tax)100 7.Calculate
totamt (subtot + taxamt)
8.Print all Details
9.Stop the Program

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
main()
{
float totamt, amount,subtot,disamt,taxamt,quantity,value,discount,tax;
clrscr();
printf("\n Enter the quantity of item sold:");
scanf("%f",&quantity);
printf("\n Enter the value of item:");
scanf("%f",&value);
printf("\n Enter the Discount percentage:");
scanf("%f",&discount);
printf("\n Enter the tax:");
scanf("%f",&tax); amount=quantity *
value;
disamt=(amount*discount)/100.0;
subtot=amount-disamt;
taxamt=(subtot*tax)/100.0;
totamt=subtot+taxamt;
printf("\n\n\n ******BILL****** ")
printf("\nQuantitysold: %f", quantity);
printf("\npriceperitem: %f", value);
printf("\nAmount: %f", amount); printf(" \n
Discount: - %f", disamt) ;
printf("\n Discounted Total: %f", subtot) ;
printf("\n Tax:=+ %f", taxamt);
printf("\n Total Amount %f", totamt);
getch();
}
```

OUTPUT:

```
 Enter the quantity of item sold:40

 Enter the value of item:20

 Enter the Discount percentage:7

 Enter the tax:7

 ******BILL******
Quantitysold: 40.000000
priceperitem: 20.000000
Amount: 800.000000
Discount: - 56.000000
Discounted Total: 744.000000
Tax:=+ 52.080002
Total Amount 796.080017
```

RESULT:

Thus the program for a given product with a VAT OF 7% to find the VAT Amount andthe total value of the product was executed and the output was verified

EX.NO:1b **DATE:**	**PROGRAM USING I/O STATEMENTS AND EXPRESSIONS**

AIM:

To write a C Program to find area and perimeter of Circle.

ALGORITHM:

```
1.Start the program.
2.Read radius.
3.Area  P1 * Radius * Radius , PI=3.14
4.Perm 2 * PI * Radius
5.Print area Prem
6.Stop.
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#define pi 3.14 void
main(){
float r,area,perimeter;
clrscr();
printf("\n Enter r value:");
scanf("%f",&r); area=pi*r*r;
printf(" area%f:",area);
perimeter=2*pi*r;
printf("perimeter%f:",perimeter);
getch();
}
```

OUTPUT:

```
Enter r value:6
area:113.040001
perimeter:37.680000
```

RESULT:

Thus the program to find area and perimeter of Circle was executed and the output was verified.

EX.NO:2 DATE:	WRITE A PROGRAM TO FIND WHETHER THE GIVEN YEAR IS LEAP YEAR OR NOT.

AIM:

To Write a program to find whether the given year is leap year or not.

ALGORITHM:

```
1.Start the program.
2.Read Year.
3.IF Year%4=0.
    Step-3.1        IF Year%100=0.
    Step-3.1.1      IF Year%400=0.
    Step-3.1.2      Print “Leap Year”.
    Step-3.1.3      ELSE Print “Not Leap Year.
    Step-3.2        ELSE Print “Leap Year”.
4.Print “Not Leap Year”.
5.Stop.
```

PROGRAM:

```
#include    <stdio.h>
#include<conio.h>
void main()
{
   int year;
   clrscr();
   printf("Enter a year: ");
   scanf("%d",&year);
   if(year%4 == 0)
   {
      if( year%100 == 0)
      {
         if ( year%400 == 0)
            printf("%d is a leap year.", year);
         else
            printf("%d is not a leap year.", year);
      }
      else
         printf("%d is a leap year.", year );
   }
   else
      printf("%d is not a leap year.", year);
   getch();
}
```

OUTPUT:

```
Enter a year: 2023
2023 is not a leap year.
```

RESULT:

Thus the program to find whether the given year is leap year or not was executed and theoutput was verified.

EX.NO:3 **DATE:**	**DESIGN A CALCULATOR TO PERFORM THE OPERATIONS NAMELY, ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION AND SQUARE OF A NUMBER.**

AIM:

Design a calculator to perform the operation, namely, addition, subtraction, multiplication, division and square of a number.

ALGORITHM:

```
1.Start the program.
2.Read a, b ,c.
3.print menu
4.read choice
5.switch(ch):
        5.1 Add
                5.1.1 Resut->a+b
                5.1.2 print result
        5.2 subract
                5.2.1 Resut->a-b
                5.2.2 print result
        5.3 multiply
                5.3.1 resut->a*b
                5.3.2 print result
        5.4 divide
                5.4.1 result->a/b
                5.4.2 print result
        5.5 square
                5.5.1 result->a*a
                5.5.2 result1->b*b
                5.5.3 print result
        6.stop
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main(){
int a,b,result,sq1,sq2,ch;
float divide;
clrscr();
printf("Enter two integers:");
scanf("%d%d",&a,&b);
printf("1.add,2.subtract,3.multiply,4.divide,5.square");

switch(ch)
{
        case 1:                         {
```

```
                        result=a+b;
                }       printf("Sum=%d\n",result);
        case 2:         break;
                {


                        result=a-b;
                }       printf("Difference=%d\n",result);break;
        case 3:
                {

                        result=a*b;
                        printf("Multiplication=%d\n",result);
                }       break;
        case 4:
                {

                        result=a/(float)b;
                        printf("Division=%.2f\n",result);
                }       break;
        case 5:
                {


                        sq1=a*a;
                        printf("Square=%d\n",sq1);
                        sq2=b*b;
                }       printf("Second square=%d\n",sq2);
                        break;

        }
        getch();
}
```

OUTPUT:

```
Enter two integers:56
10
1.add,2.subtract,3.multiply,4.divide,5.square
Enter the choice;1
Sum=66
```

RESULT:

Thus the program to design a calculator to perform the operation namely, addition, subtraction, multiplication, division and square of a number was executed and the output wasverified.

EX.NO:4 **DATE:**	**CHECK WHETHER A GIVEN NUMBER IS ARMSTRONG NUMBER OR NOT?**

AIM:

To write a C Program to Check whether a given number is Armstrong number or not.

ALGORITHM:

1. Start
2. Declare variables
3. Read the Input number.
4. Calculate sum of cubic of individual digits of the input.
5. Match the result with input number.
6. If match, Display the given number is Armstrong otherwise not.
7. Stop

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int number, sum = 0, rem = 0, cube = 0, temp;
clrscr();
printf ("enter a number");
scanf("%d", &number);
temp = number;
while (number != 0)
{
rem = number % 10;
cube = pow(rem, 3);
sum = sum + cube;
number = number / 10;
}
if (sum == temp)
printf ("The given no is armstrong no");else
printf ("The given no is not a armstrong no");
getch();
}
```

OUTPUT:

```
enter a number:21
The given nnumber is not a armstrong number
```

RESULT:

Thus a C Program for Armstrong number checking was executed and the output was obtained.

EX.NO:5 DATE:	SORT THE NUMBERS BASED ON THE WEIGHT

AIM:

To write a C Program to perform the following: Given a set of numbers like <10, 36, 54, 89, 12, 27>, find sum of weights based on the following conditions

- 5 if it is a perfect cube
- 4 if it is a multiple of 4 and divisible by 6
- 3 if it is a prime number

ALGORITHM:

1. Start
2. Declare variables
3. Read the number of elements .
4. Get the individual elements.
5. Calculate the weight for each element by the conditions
   - 5 if it is a perfect cube (pow)
   - 4 if it is a multiple of 4 and divisible by 6 (modulus operator)
   - 3 if it is a prime number(modulus operator)
6. Display the output of the weight calculations after sorting .
7. Stop

PROGRAM:

```
#include    <stdio.h>
#include<conio.h>
#include    <math.h>
void main()
{
int nArray[50],wArray[50],nelem,i,j,t;
clrscr();
printf("\nEnter the number of elements in an array : ");
scanf("%d",&nelem);
printf("\nEnter %d elements\n",nelem);
for(i=0;i<nelem;i++)
scanf("%d",&nArray[i]);
//Calculate the weight
for(i=0; i<nelem; i++)


{
wArray[i] = 0;
if(percube(nArray[i]))
wArray[i] = wArray[i] + 5;
```

```
    if(nArray[i]%4==0 && nArray[i]%6==0)
    wArray[i] = wArray[i] + 4; if(prime(nArray[i]))
    wArray[i] = wArray[i] + 3;
    }
    // Sorting an array
    for(i=0;i<nelem;i++)
    for(j=i+1;j<nelem;j++)
    if(wArray[i] > wArray[j])
    {
    t = wArray[i]; wArray[i] =
    wArray[j];wArray[j] = t;
    }
    for(i=0; i<nelem; i++) printf("<%d,%d>\n",
    nArray[i],wArray[i]);getch();
    }
    int prime(int num)

    {
    int flag=1,i;
    for(i=2;i<=num/2;i++)
    if(num%i==0)
    {
    flag=0;
    break;
    }
    return flag;
    }
    int percube(int num)
    {
    int i,flag=0;
    for(i=2;i<=num/2;i++)
    if((i*i*i)==num)
    {
    flag=1;
    break;
    }
    return flag;

}
```

OUTPUT:

```
Enter the number of elements in an array : 5

Enter 5 elements
1
2
3
4
5
<1,7>
<2,0>
<3,0>
<4,0>
<5,1811939693>
```

RESULT:

Thus a C Program for Sort the numbers based on the weight was executed and the outputwas obtained.

EX.NO:6 DATE:	SORTING NUMBERS BASED ON THEIR WEIGHT IN INCREASING ORDER

AIM:

Write a program for Sort the numbers based on the weight in the increasing order as shown below <10, its weight <36, its weight> <89, its weight>

ALGORITHM:

1. Start.
2. Declare variables
3. Read the number of elements.
4. Get the individual elements.
5. Calculate the weight for each element by the conditions
    - 5 if it is a perfect cube (pow)
    - 4 if it is a multiple of 4 and divisible by 6 (modulus operator)
    - 3 if it is a prime number (modulus operator)
6. Display the output of the weight calculations after sorting
7. Stop

PROGRAM:

```
#include <stdio.h>

int getWeight(int num) {
    int weight = 0;
    while (num > 0) {
        weight += num % 10;
        num /= 10;
    }
    return weight;
}

// function to compare the weights of two numbers
int compare(int num1, int num2) {
    int weight1 = getWeight(num1);
    int weight2 = getWeight(num2);
    if (weight1 < weight2)
        return -1;



    else if (weight1 > weight2)
        return 1;
    else
        return 0;
}

// bubble sort to sort the numbers based on their weights
void sort(int arr[], int n) {
    int i, j, temp;
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (compare(arr[j], arr[j+1]) > 0) {
                // swap the numbers
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

// main function
int main() {
    int arr[] = {10,89,36};
    int n = sizeof(arr) / sizeof(arr[0]);
    sort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

OUTPUT:

```
Sorted array: 10 36 89
```

RESULT:

Thus a C Program sorting numbers based on their weight in increasing order was executed and the output was obtained**.**

EX.NO:7

DATE:

AVERAGE HEIGHT OF PERSONS

AIM:

To write a C Program to populate an array with height of persons and find how many persons are above the average height.

ALGORITHM:

1. Start
2. Declare variables
3. Read the total number of persons and their height.
4. Calculate avg=sum/n and find number of persons their h>avg.
5. Display the output of the calculations.
6. Stop.

PROGRAM:

```
/* Get a Height of Different Persons and find how many of them are are above average */#include
<stdio.h>
#include <conio.h>
void main()
{
int i,n,sum=0,count=0,height[100];
float avg;
clrscr();
printf("Enter the Number of Persons : ");
scanf("%d",&n);
printf("\n Enter the Height of each person in centimeter\n");
for(i=0;i<n;i++)
{
scanf("%d",&height[i]);
sum = sum + height[i];
}
avg = (float)sum/n;
//Counting
for(i=0;i<n;i++)
if(height[i]>avg)
count++;
printf("\n Average Height of %d persons is : %.2f\n",n,avg); printf("\n
The number of persons above average : %d ",count);getch();

}
```

OUTPUT:

```
Enter the Number of Persons: 4

 Enter the Height of each person in centimeter
```

```
200
280
220
270

 Average Height of 4 persons is: 242.50

 The number of persons above average: 0
```

RESULT:

Thus a C Program average height of persons was executed and the output was obtained.

EX.NO:8

DATE:

STRING OPERATIONS

AIM:

To write a C Program to perform string operations on a given paragraph for the following using built-in functions:

a. Find the total number of words.

b. Capitalize the first word of each sentence.

c. Replace a given word with another word.

ALGORITHM:

1. Start
2. Declare variables
3. Read the text.
4. Display the menu options
5. Compare each character with tab char „\t‟ or space char „ „ to count no of words
6. Find the first word of each sentence to capitalize by checks to see if a character is a punctuation mark used to denote the end of a sentence. (! . ?)
7. Replace the word in the text by user specific word if match.
8. Display the output of the calculations.
9. Repeat the step 4 till choose the option stop.
10. Stop.

PROGRAM:

A. Find the total number of words.

```
#include<stdio.h>
#include<conio.h>
#define MAX_SIZE 100 // Maximum string sizevoid
main()
{
    char str[MAX_SIZE];
    char prevChar;
    int i, words;
    clrscr();
    /* Input string from user */
    printf("\nEnter any string: ");
    gets(str);i
    = 0;
    words = 0;
    prevChar = '\0'; // The previous character of str[0] is null
    /* Runs loop infinite times */
    while(1)
    {
        if(str[i]==' ' || str[i]=='\n' || str[i]=='\t' || str[i]=='\0')
        {
            /**
             * It is a word if current character is whitespace and
             * previous character is non-white space.
             */
            if(prevChar != ' ' && prevChar != '\n' && prevChar != '\t' && prevChar != '\0')
            {
                words++;
            }
        }
        /* Make the current character as previous character */
        prevChar = str[i];

        if(str[i] == '\0')
             break;
        else
             i++;
    }
    printf("Total number of words = %d", words);
    getch();
}
```

OUTPUT:

```
Enter any string: c programming
Total number of words = 2
```

b. Capitalize the first word of each sentence.

```
#include<stdio.h>
#include<conio.h>
#define MAX 100
void main()


{
char str[MAX]={0};int i;
clrscr();
//input string printf("Enter a
string: ");
scanf("%[^\n]s",str); //read string with spaces
//capitalize first character of words
for(i=0; str[i]!='\0'; i++)
{
//check first character is lowercase alphabet
if(i==0)
{
if((str[i]>='a' && str[i]<='z'))
str[i]=str[i]-32; //subtract 32 to make it capital
continue; //continue to the loop
}
if(str[i]==' ')//check space
{
//if space is found, check next character
++i;
//check next character is lowercase alphabet

if(str[i]>='a' && str[i]<='z')
{
str[i]=str[i]-32; //subtract 32 to make it capital
continue; //continue to the loop
}
}
else
{
//all other uppercase characters should be in lowercase
if(str[i]>='A' && str[i]<='Z')
str[i]=str[i]+32; //subtract 32 to make it small/lowercase
}
}
printf("Capitalize string is: %s\n",str);
getch();
}
```

OUTPUT:

```
Enter a string: c programming lab
Capitalize string is: C Programming Lab
```

c. Replace a given word with another word.

```
#include <stdio.h>
#include <string.h>
#include<conio.h>
void main()
{
  char s[] = "All work and no play makes Jack a dull boy.";char
  word[10],rpwrd[10],str[10][10];
  int i=0,j=0,k=0,w,p;
  printf("All work and no play makes Jack a dull boy.\n");
  printf("\nENTER WHICH WORD IS TO BE REPLACED\n");
  scanf("%s",word);
  printf("\nENTER BY WHICH WORD THE %s IS TO BE REPLACED\n",word);
  scanf("%s",rpwrd);
  p=strlen(s);
  for (k=0; k<p; k++)
    {
      if (s[k]!=' ')
        {
          str[i][j] = s[k];
          j++;
        }
      else
        {
          str[i][j]='\0';
          j=0; i++;
        }
    }
  str[i][j]='\0';
  w=i;
  for (i=0; i<=w; i++)
    {
      if(strcmp(str[i],word)==0)
      strcpy(str[i],rpwrd);
      printf("%s ",str[i]);
    }

  getch();
}
```

OUTPUT:

```
All work and no play makes Jack a dull boy.

ENTER WHICH WORD IS TO BE REPLACED
boy

ENTER BY WHICH WORD THE boy IS TO BE REPLACED
girl
All work and no play makes Jack a dull boy.
```

RESULT:

Thus a C Program String operations was executed and the output was obtained

EX.NO:9 DATE:	TOWERS OF HANOI USING RECURSION

AIM:

To write a C Program to Solve towers of Hanoi using recursion.

ALGORITHM:

1. Start
2. Declare variables
3. Read the Input for number of discs.
4. Check the condition for each transfer of discs using recursion.
5. Display the output of the each move .
6. Stop

PROGRAM:

```
#include <stdio.h>
void towers(int, char, char, char);
void main()
{
    int num;
clrscr();
    printf("Enter the number of disks : ");
    scanf("%d", &num);
    printf("The sequence of moves involved in the Tower of Hanoi are :\n");
    towers(num, 'A', 'C', 'B');
    getch();
}
void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}
```

OUTPUT:

```
Enter the number of disks : 4
The sequence of moves involved in the Tower of Hanoi are :
```

```
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 3 from peg A to peg B
Move disk 1 from peg C to peg A
Move disk 2 from peg C to peg B
Move disk 1 from peg A to peg B
Move disk 4 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 2 from peg B to peg A
Move disk 1 from peg C to peg A
Move disk 3 from peg B to peg C
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
```

RESULT:

Thus a C Program Towers of Hanoi using Recursion was executed and the output was obtained

EX.NO:10 DATE:	LOCATE AND DISPLAY THE CONTENTS OF AN ARRAY USING POINTERS

AIM:

To write a program for locate and display the contents of an array using pointers

ALGORITHM:

1. Start
2. Define an array of the desired data type and size.
3. Declare a pointer variable of the same data type as the array.
4. Assign the address of the first element of the array to the pointer variable.
5. Loop through the array using the pointer variable and print the contents of each element.
6. Stop

PROGRAM:

```
#include <stdio.h>

int main() {
   int arr[5] = {10, 20, 30, 40, 50};
   int *ptr = arr; // pointer to the first element of the array

   printf("Array contents: ");
   for (int i = 0; i < 5; i++) {
      printf("%d ", *ptr); // print the contents of the current element
      ptr++; // move the pointer to the next element
   }
   return 0;
}
```

OUTPUT:

```
Array contents: 10 20 30 40 50
```

RESULT:

Thus a C Program locate and display the contents of an array using pointers was executed and the output was obtained

EX.NO:11 **DATE:**	**GENERATE SALARY SLIP OF EMPLOYEES USING STRUCTURES AND POINTERS**

AIM:

To write a C Program to Generate salary slip of employees using structures and pointers.

ALGORITHM:

1. Start
2. Declare variables
3. Read the number of employees .
4. Read allowances, deductions and basic for each employee.
5. Calculate net pay= (basic+ allowances)-deductions
6. Display the output of the Pay slip calculations for each employee.
7. Stop

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
struct emp
{
 int empno ; char
name[10] ;
nt bpay, allow, ded, npay ;
} e[10] ;
void main()
{
    int i, n ; clrscr()
    ;
    printf("Enter the number of employees : ") ;
    scanf("%d", &n) ;
    for(i = 0 ; i < n ; i++)
    {
        printf("\nEnter the employee number : ");
        scanf("%d", &e[i].empno) ;
        printf("\nEnter the name : ") ;
        scanf("%s", e[i].name) ;
        printf("\nEnter the basic pay, allowances & deductions : ") ;
        scanf("%d %d %d", &e[i].bpay, &e[i].allow, &e[i].ded) ;
        e[i].npay = e[i].bpay + e[i].allow - e[i].ded ;


    }
printf("\nEmp. No. Name \t Bpay \t Allow \t Ded \t Npay \n\n") ;for(i = 0
; i < n ; i++)
{
printf("%d \t %s \t %d \t %d \t %d \t %d \n", e[i].empno,e[i].name, e[i].bpay, e[i].allow, e[i].ded,e[i].npay) ;
}
    getch() ;
}
```

OUTPUT:

```
Enter the number of employees : 2

Enter the employee number : 1

Enter the name : sam

Enter the basic pay, allowances & deductions : 3000
200
100

Enter the employee number : 2

Enter the name : Albert

Enter the basic pay, allowances & deductions : 4000
200
100

Emp. No. Name    Bpay    Allow   Ded     Npay

1       sam            3000    200     100     3100
2       Albert         4000    200     100     4100
```

RESULT:

Thus a C Program Salary slip of employees was executed and the output was obtained.

EX.NO:12 DATE:	BANKING APPLICATION

AIM:

To write a C Program to Count the number of account holders whose balance is less than the minimum balance using sequential access file.

ALGORITHM:

1. Start
2. Declare variables and file pointer.
3. Display the menu options.
4. Read the Input for transaction processing.
5. Check the validation for the input data.
6. Display the output of the calculations .
7. Repeat step 3 until choose to stop.
8. Stop

PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#define MINBAL 500
struct Bank_Account
{
  char no[10];
  char name[20];
  char balance[15];
};
struct Bank_Account acc;
void main()
{
   long int pos1,pos2,pos;
   FILE *fp;
   char *ano,*amt;
   char choice;
   int type,flag=0;
   float bal;
   do{
     clrscr();
     fflush(stdin);
     printf("1. Add a New Account Holder\n");printf("2.
     Display\n");
     printf("3. Deposit or Withdraw\n");
     printf("4. Number of Account Holder Whose Balance is less than the Minimum Balance\n");printf("5.
     Delete All\n");

     printf("6. Stop\n"); printf("Enter
     your choice : ");
     choice=getchar();
  switch(choice)
  {
    case '1' :
 fflush(stdin);
 fp=fopen("acc.dat","a");
 printf("\nEnter the Account Number : ");
 gets(acc.no);
 printf("\nEnter the Account Holder Name : ");
 gets(acc.name);
 printf("\nEnter the Initial Amount to deposit : ");
 gets(acc.balance);
 fseek(fp,0,2);
 fwrite(&acc,sizeof(acc),1,fp);
 fclose(fp);
 break;
    case '2' :
 fp=fopen("acc.dat","r");
 if(fp==NULL) printf("\nFile is
 Empty"); else
 {
 printf("\nA/c Number\tA/c Holder Name Balance\n");
 while(fread(&acc,sizeof(acc),1,fp)==1)
    printf("%-10s\t\t%-20s\t%s\n",acc.no,acc.name,acc.balance);
 fclose(fp);
 }
 break;
    case '3' :
 fflush(stdin);
 flag=0;
 fp=fopen("acc.dat","r+"); printf("\nEnter the
 Account Number : ");gets(ano);
 for(pos1=ftell(fp);fread(&acc,sizeof(acc),1,fp)==1;pos1=ftell(fp))
 {
   if(strcmp(acc.no,ano)==0)
   {
     printf("\nEnter the Type 1 for deposit  & 2 for withdraw : ");
     scanf("%d",&type);
     printf("\nYour Current Balance is : %s",acc.balance);
     printf("\nEnter the Amount to transact : "); fflush(stdin);
     gets(amt);
     if(type==1)
bal = atof(acc.balance) + atof(amt);



else
     {
bal = atof(acc.balance) - atof(amt);
if(bal<0)
{
  printf("\nRs.%s Not available in your A/c\n",amt);
  flag=2;
  break;
}
     }
     flag++;
     break;
```

```
           }


          }
          if(flag==1)
          {
             pos2=ftell(fp); pos
             = pos2-pos1;
             fseek(fp,-pos,1);
             sprintf(amt,"%.2f",bal);
             strcpy(acc.balance,amt);
             fwrite(&acc,sizeof(acc),1,fp);
          }
          else if(flag==0)
            printf("\nA/c Number Not exits... Check it again");
          fclose(fp);
          break;

             case '4' :
        fp=fopen("acc.dat","r");
        flag=0;
        while(fread(&acc,sizeof(acc),1,fp)==1)
        {
           bal = atof(acc.balance);
           if(bal<MINBAL) flag++;
        }
        printf("\nThe Number of Account Holder whose Balance less than the Minimum Balance :
      %d",flag);
        fclose(fp);
        break;
             case '5' :
        remove("acc.dat");
        break;
             case '6' :
        fclose(fp);
        exit(0);


           }
          printf("\nPress any key to continue.........");
          getch();
       } while (choice!='6');
       }
```

OUTPUT:

```
1. Add a New Account Holder
2. Display
3. Deposit or Withdraw
4. Number of Account Holder Whose Balance is less than the Minimum Balance
5. Delete All
6. Stop
Enter your choice : 1

Enter the Account Number : 2314325
Enter the Account Holder Name : NEW ACC

Enter the Initial Amount to deposit : 234

Press any key to continue....1. Add a New Account Holder
2. Display
3. Deposit or Withdraw
4. Number of Account Holder Whose Balance is less than the Minimum Balance
5. Delete All
6. Stop
Enter your choice : 2

A/c Number        A/c Holder Name     Balance
123456789             ALBERT              1000
123456789             SUZEN                2000
123456789             MARRY               1000
123456789             JHON                  2000
987654321             WILLIAM            20000
2314325               ZACK                 23400
```

RESULT:

Thus a C Program for Banking Application was executed and the output was obtained.

EX.NO:13

DATE:

GENERATE A FIBONACCI SERIES

AIM:

To write a c program to generate a Fibonacci series

ALGORITHM:

```
1. Start
2. Declare variables i, a,b , show
3. Initialize the variables, a=0, b=1, and show =0
4. Enter the number of terms of Fibonacci series to be printed
5. Print First two terms of series
6. Use loop for the following steps
7. show=a+b
         a=b
         b=show
        print the value of show
8. end
```

PROGRAM:

```
#include <stdio.h>

int main()

{

   int i, n, a=0, b=1, show=0;

   printf("\nEnter number of terms required in Fibonacci Series: ");

   scanf("%d",&n);

   printf("\nThe Fibonacci Series is:\n\n\n %d+%d+", a, b);

   i=2;

   while (i<n)

   {

      show=a+b;

      a=b;

      b=show;

      ++i;

      printf("%d+",show);
```

```
    }
   return 0;
}
```

OUTPUT:

```
Enter number of terms required in Fibonacci Series: 20

The Fibonacci Series is:

0+1+1+2+3+5+8+13+21+34+55+89+144+233+377+610+987+1597+2584+4181+
```

RESULT:

Thus a C Program for generate a Fibonacci was executed and the output was obtained**.**

EX.NO:14

DATE:

PALINDROME PROGRAM IN C

AIM:

To write a palindrome program in c

ALGORITHM:

1. Start
2. Declare and initialize the necessary variables
3. Enter the number to be checked for palindrome property
4. Assign the number with a temporary variable
5. Reverse the number using loop
6. Store the reversed number
7. Compare the reversed number and original number stored in a temporary variable
8. If the number and its reverse are found to be the same, print the number as a palindrome number
9. Otherwise, the number is not a palindrome
10. Stop

PROGRAM:

```
#include <stdio.h>
int main()
{
   int num, rev = 0, temp; // declaration of variable
   printf(" Enter a number: ");// Asking for number to check
   scanf("%d",&num);
   temp = num; // Equalizing of temporary variable to the number
   while( temp != 0 ) // loop to reverse the number
   {
     rev = rev * 10;
     rev = rev + temp%10;
     temp = temp/10;
   }


   if ( num == rev ) // comparison of reverse with the original number
      printf(" %d is a Palindrome Number.\num", num);
   else
```

```
        printf(" %d is not a Palindrome Number.\num", num);

    return 0;
}
```

OUTPUT:

```
Enter a number: 121

121 is a Palindrome Number.
```

RESULT:

Thus a C Program for a palindrome was executed and the output was obtained.