

C PROGRAMMING USING SIMPLE STATEMENTS AND EXPRESSIONS**EX.NO: 1****EVALUATE THE GIVEN EXPRESSION****AIM:**

To write a C program to evaluate the given expression

ALGORITHM:

Step-1 Start the program.

Step-2 Input the values for declared variables

Step-3 Substitute the values in expression and calculate the results.

Step-4 Print the results

Step-5 Stop

PROGRAM:

```
/* Expression Evaluation*/  
  
#include<stdio.h>  
main()  
{  
    int a,b,c,d;  
    float x,y,z;  
    printf ("Enter the values for a,b,c,d\n");  
    scanf("%d%d%d%d", &a,&b,&c,&d);  
    x = (a * b) - c;  
    y = (b/c) * a;  
    z = (a - b) / (c + d)  
    printf(" The value of x is %f ",x );  
    printf("The value of y is %f ",y );  
    printf("The value of z is %f ",z );  
  
}
```

OUTPUT:

Enter the value for a,b,c,d

a= 5; b = 6; c= 12;d=2

x = 28.000; y =

2.5000 z = 0.0555

RESULT:

Thus the program to evaluate the given expression was performed and executed successfully.

EX.NO: 2**DECISION MAKING****AIM:**

To write a c program to check whether given Number is odd or even.

ALGORITHM:

- 1: Declare a variable to get a Number
- 2: Read the input
- 3: Get the remainder of given number using modulo operator
- 4: If remainder is 0 prints "Even Number", else print "Odd Number".

PROGRAM:

```
#include<stdio.h>
main()
{
int a,rem;
printf("Enter a Number\n");
scanf("%d",&a);
rem=a%2;
if(rem==0)
printf("The Given Number is Even"); else
printf("The Given Number is Odd");
}
```

OUTPUT:

```
Enter a Number 13
The Given Number is Odd
```

RESULT:

Thus the program to check whether given Number is odd or even was performed and executed successfully.

EX.NO:3**THE GIVEN YEAR IS LEAP YEAR OR NOT****AIM:**

To write a c program to check whether the given year is leap year or not.

ALGORITHM:

- Step 1: Start the Program
- Step 2: Enter the year
- Step 3: check the given year is divisible by 4, it denotes leap year.
- Step 4: Otherwise it is not leap year.
- Step 5: print the result
- Step 7: Stop

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int year;
clrscr();
printf("enter the year\n");
scanf("%d",&year);
if(year%4==0 &&year%100 !=0|| year%400==0)
printf("leap year");
else
printf("not leap year");
getch();
}
```

OUTPUT:

```
Enter the year = 2016
Leap year
```

RESULT:

Thus the C program for the given year is leap year or not was performed and executed successfully.

EX.NO:4 DEMONSTRATE ARITHMETIC OPERATIONS (USING SWITCH...CASE)**AIM:**

To write a C program for demonstrating arithmetic operations using switch case statement.

ALGORITHM:

Step1: Start the program

Step 2: Display menu showing addition, subtraction, multiplication and division operation.

Step-3: Get the values for two variables

Step 4: Obtain the choice from the user and accordingly switch over to particular block.

Step 5: Display the result.

Step 6: If the user wishes to continue repeat steps 2 and3

Step 7: Stop

PROGRAM:

```
/* Program to demonstrate arithmetic operations */
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b, c, n;
    clrscr();
    printf("1. Addition\n");
    printf("2. Subtraction\n");
    printf("3. Multiplication\n");
    printf("4. Division\n");
    printf("5. Square of a No\n");
    printf("0. Exit\n");
    printf("Enter your choice : ");
    scanf("%d",&n);
    printf("Enter the two numbers :");
    scanf("%d%d",&a,&b);
    switch(n)
    {
    case 1:
        c = a + b;
        printf("Addition :%d\n",c);
        break;
    case 2:
        c = a - b;
        printf("Subtraction :%d\n",c);
        break;
    case 3:
        c = a * b;
        printf("Multiplication :%d\n",c);
```

```
        break;
    case 4:
        c = a / b;
        printf("Division :%d\n",c);
        break;
    case 5:
        c=a*a;
        printf("Square : %d\n",a);
    case 0:
        exit(0);
        break;
    }
    getch();
}
```

OUTPUT:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Square
0. Exit

Enter Your Choice : 1
Enter the 2 nos a and b: 2 8
Addition : 10.

Enter Your Choice : 2

Enter the 2 nos a and b: 5 2
Subtraction : 3.

Enter Your Choice : 3
Enter the 2 nos a and b: 2 8
Multiplication : 16.

Enter Your Choice : 4.
Enter the 2 nos a and b: 8 4
Division : 2.

Enter your Choice: 5
Enter the number a : 2
Square : 4

Enter Your Choice : 0.
Exit.

RESULT:

Thus the C program for demonstrating arithmetic operations using switch case statement was performed and executed successfully.

EX.NO: 5**ARMSTRONG NUMBER****AIM:**

Write a C program to check whether the given number is Armstrong number or not.

ALGORITHM:

Step-1 Start the program

Step-2 Enter the number up to which Armstrong numbers are to be generated.

Step-3 Set a loop upto the number

Step-4 Sum the cube of each individual digit of the number and store the sum in s.

Step-5 Check whether the entered digit and calculated sum are equal.

Step-6 Find whether the reverse number is equal to the given number. If equal the number is Armstrong else not an Armstrong number

Step-7 Increment the loop and perform steps 5 and 6 till the end of loop is reached.

Step-8 Stop.

PROGRAM:

```
#include <stdio.h>
int main()
{
    int number, originalNumber, remainder, result = 0;

    printf("Enter a three digit integer: ");
    scanf("%d", &number);

    originalNumber = number;

    while (originalNumber != 0)
    {
        remainder = originalNumber%10;
        result += remainder*remainder*remainder;
        originalNumber /= 10;
    }

    if(result == number)
        printf("%d is an Armstrong number.",number);
    else
        printf("%d is not an Armstrong number.",number);
}
```

OUTPUT:

Enter a three digit integer: 371
371 is an Armstrong number.

RESULT:

Thus the program to check whether the given number is present or not was performed and executed successfully.

EX.NO: 6**SORT THE NUMBERS BASED ON WEIGHT****AIM:**

To write a C program to sort the numbers based on weight in the increasing order.

ALGORITHM:

Step -1 Read the weight of a person

Step-2 Check whether the given weight is perfect square of 5

Step-3 Check whether the weight is divisible by 3 and 4

Step-4 Print the weight is increasing order

Step-5 stop

PROGRAM:

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

int getWeight(int n)
{
int w=0;
float root=sqrt(n);
if(root==ceil(root))
    w+=5;
if(n%4==0&& n%6==0)
    w+=4;
if(n%2==0)
    w+=3;
return w;
}

void main()
{
int nums[15];
int ws[15];
int i,j,t,n;
printf("Enter number of numbers");
scanf("%d",&n);
printf("\nEnter numbers");
for(i=0;i<n;i++)
    scanf("%d",&nums[i]);
for(i=0;i<n;i++)
    ws[i]=getWeight(nums[i]);
printf("\nBefore sorting:\n");
for(i=0;i<n;i++)
    printf("%d:%d\t",nums[i],ws[i]);
for(i=0;i<n;i++)
    for(j=0;j<n-i-1;j++)
        if(ws[j]>ws[j+1])
            {
            t=ws[j+1];
            ws[j+1]=ws[j];
            ws[j]=t;
            t=nums[j+1];
            nums[j+1]=nums[j];
            nums[j]=t;
            }
```

8

```
    }  
    printf("\nSorted:\n");  
    for(i=0;i<n;i++)  
        printf("%d:%d\t",nums[i],ws[i]);  
}
```

OUTPUT:

Enter the Number of Numbers

3

Enter the Number

34

56

25

Sorted

25

34

56

RESULT:

Thus the program to sort the numbers based on weight in the increasing order was performed and executed successfully.

EX.NO:7**HEIGHT AND WEIGHT OF THE PERSONS****AIM:**

To write a C program to check the numbers of persons height, weight and average height

ALGORITHM:

Step-1 Take the height of a person as input and store it in the variable height.

Step-2 If the variable height is lesser than 150 cm, then print the output as “Dwarf”.

Step-3 If the variable height is lesser than or equal to 165 cm and greater than or equal to 150 cm, then print the output as “Average Height”.

Step-4 If the variable height is lesser than or equal to 195 cm and greater than 165 cm, then print the output as “Taller”.

Step-5 Stop.

PROGRAM :

```
#include <stdio.h>
void main()
{
float height;
printf("Enter the height (in centimeters) \n");
scanf("%f",&height );
if (height < 150.0)
printf("Dwarf\n");
else if ((height>=150.0)&& (height <=165.0)
printf("average height \n");
else if ((height > 165.0) && (height <=195.0))
printf("Taller \n");
else
printf("Abnormal height ");
}
```

OUTPUT:

```
Enter the Height (in centimetres)
165
Average Height
Enter the Height (in centimetres)
140
Dwarf
Enter the Height (in centimetres)
190
Taller
```

RESULT:

Thus the C program to check the numbers of persons height, weight and average height was performed and executed successfully.

EX.NO:8 COMPUTE THE BODY MASS INDEX OF THE INDIVIDUALS**AIM:**

Write a C program with height and weight of persons and compute the body mass index of the individuals.

ALGORITHM:

Step-1 : Read the values of weight and Height
Step-2 : Calculate BMI=Weight/Height/Height
Step-3 : Print the value of BMI
Step-4: Stop

PROGRAM:

```
/* Calculation of BMI (Body Mass Index) */  
#include <stdio.h>  
double bmi_cal(double h, double w) {  
    // BMI = weight(kg) / height(m) / height(m)  
    return w/h/h;  
}  
int main(void) {  
    double height, weight, bmi;  
    printf("Input height(m) and weight(kg) ");  
    scanf("%lf%lf", &height, &weight);  
    // Calculation of BMI  
    bmi = bmi_cal(height, weight);  
    printf("%5.2f m, %6.2f kg: BMI = %6.2f\n", height, weight, bmi);  
    return (0);  
}
```

OUTPUT:

Input height(m) and weight(kg) 1.65 70

1.65 m, 70.00 kg: BMI = 25.71

RESULT:

Thus the program for body mass index was executed successfully.

EX.NO:9**STRING REVERSE****AIM:**

Write a C program to print the string reverse.

ALGORITHM:

Step-1 Traverse the input string with two variables l and r where l =0 and r = n-1, n is the length of the input string

Step-2 while l < r

Step-3 if s[l] is not a alphabet, then do l++

Step-4 else if s[r] is not a alphabet, then do r--

Step-5 else swap the characters in s[l] and s[r] and so l++, r--

PROGRAM :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char str[10],temp;
    int i,len;
    printf("Enter String : ");
    scanf("%s",str);
    len=strlen(str)-1;
    for(i=0;i<strlen(str)/2;i++)
    {
        temp=str[i];
        str[i]=str[len];
        str[len--]=temp;
    }
    printf("%s",str);
    getch();
}
```

OUTPUT:

```
Enter String
madam
madam
```

RESULT:

Thus the program for string reverse was executed successfully.


```
        break;
    case 2: printf("\nOctal Value:\t");
            conversion(num, 8);
            break;
    case 3: printf("\nHexadecimal Value:\t");
            conversion(num, 16);
            break;
    case 4: exit(0);
    default: printf("Enter a correct input\n");
}
}
printf("\n");
return 0;
}
```

OUTPUT:

Enter a decimal integer

134

Input number is = 134

Its binary equivalent is = 10000110

No.of 1's in the binary number is = 3

RESULT:

Thus the program for decimal to binary, octal and hexadecimal was executed successfully.

EX.NO:11**BUILT IN FUNCTIONS****AIM:**

Write a C program for the following

- a) Find the total number of words
- b) Capitalize the first word of each sentence
- c) Replace a given word with another word

ALGORITHM:

Step-1. Take a string as input and store it in the array s[].

Step-2. Using for loop search for a space ' ' in the string and consecutively increment a variable count.

Step-3. Do step-2 until the end of the string.

Step-4. Increment the variable count by 1 and then print the variable count as output.

```

/*
 * C Program to Count Number of Words in a given Text Or Sentence
 */
#include <stdio.h>
#include <string.h>

void main()
{
    char s[200];
    int count = 0, i;

    printf("enter the string\n");
    scanf("%[^\n]s", s);
    for (i = 0; s[i] != '\0'; i++)
    {
        if (s[i] == ' ')
            count++;
    }
    printf("number of words in given string are: %d\n", count + 1);
}
.

```

OUTPUT:

```

enter the string
welcome to sanfoundry's c-programming class!
number of words in given string are: 5

```

Program to capitalize first character of each word in a string in C

```

#include <stdio.h>
#define MAX 100

```

```

int main()
{

```

```

char str[MAX]={0};
int i;

//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);

return 0;
}

```

OUTPUT:

First run:

Enter a string: HELLO FRIENDS HOW ARE YOU?

Capitalize string is: Hello Friends How Are You?

Second run:

Enter a string: hello friends how are you?

Capitalize string is: Hello Friends How Are You?

Third run:

Enter a string: 10 ways to learn programming.

Capitalize string is: 10 Ways To Learn Programming.

Replace a word

```
#include<stdio.h>
#include<string.h>
main()
{
int lr,m,x,t,l,ll,ls,i,j,k,flag,count=0;
char line[200],str[20],rep[20];
printf("Enter line of text\n");
gets(line);
printf("Enter the word to replace\n");
scanf("%s",str);
printf("Enter the word to replace with\n");
scanf("%s",rep);
ll=strlen(line);
ls=strlen(str);
lr=strlen(rep);
for(i=0;i<ll;i++)
{
if(line[i]==str[0]&&((line[i-1]==' '||i==0)&&(line[i+ls]==' '||line[i+ls]==')))
{
for(flag=0,k=i,j=0;j{
if(line[k]==str[j])
{
flag++;
}
}
}
if(flag==ls)
{
if(lr>ls)
{
for(m=lr-ls;m>0;m--)
{
ll=strlen(line);
for(l=ll;l>i;l-)
{
line[l+1]=line[l];
}
}
}
else if(lr{
for(m=ls-lr;m>0;m--)
{
ll=strlen(line);
for(l=i;l<ll;l++)
{
line[l]=line[l+1];
```



```
}  
}  
}  
else  
{  
}  
for(x=0,t=i;x<l;r;x++,t++)  
{  
line[t]=rep[x];  
}  
}  
}  
}  
printf("Text\n");  
puts(line);  
}
```

OUTPUT :

Enter line of text
Snehal IS A GOOD GIRL
Enter the word to replace
GIRL
Enter the word to replace with
BOY
Text
Snehal IS A GOOD BOY

RESULT:

Thus the program for a)total number of words b)Capitalize each word c)replace each word were executed successfully.

EX.NO:12**TOWERS OF HANOI****AIM:**

Write a C program to solve towers of Hanoi using recursion.

ALGORITHM:

Step-1: Read the number of Disks

Step-2: Move the third disk in to empty tower

Step-3: Move Second disk into second tower

Step-4: Maintain first disk into same tower

Step-5: Move Second disk to first tower

Step-6: Move third disk in to first tower

PROGRAM:

```
/* C program for Tower of Hanoi using Recursion */

#include <stdio.h>

void towers(int, char, char, char);

int main()
{
    int num;
    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');
    return 0;
}

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 : 3

The sequence of moves involved in the Tower of Hanoi are :

Move disk 1 from peg A to peg C

Move disk 2 from peg A to peg B

Move disk 1 from peg C to peg B

Move disk 3 from peg A to peg C

Move disk 1 from peg B to peg A

Move disk 2 from peg B to peg C

Move disk 1 from peg A to peg C

RESULT:

Thus the program for Towers of Hanoi problem was executed successfully.

EX.NO:13 SORT THE LIST OF NUMBERS USING PASS BY REFERENCE**AIM:**

Write a C program to sort the list of numbers using pass by reference.

ALGORITHM:

Step-1 : Read the list of numbers

Step-2 : Call the function sort

Step-3: Sort the values using temporary variable

Step-4: Print the values

Step-5 : Stop

PROGRAM:

```
void sort(int m, int x[ ]);
main()
{
    int i;
    int marks[5] = {40, 90, 73, 81, 35};

    printf("Marks before sorting\n");
    for(i = 0; i < 5; i++)
        printf("%d ", marks[i]);
    printf("\n\n");

    sort (5, marks);
    printf("Marks after sorting\n");
    for(i = 0; i < 5; i++)
        printf("%4d", marks[i]);
    printf("\n");
}
void sort(int m, int x[ ])
{
    int i, j, t;

    for(i = 1; i <= m-1; i++)
        for(j = 1; j <= m-i; j++)
            if(x[j-1] >= x[j])
                {
                    t = x[j-1];
                    x[j-1] = x[j];
                    x[j] = t;
                }
}
```

OUTPUT:

Marks before sorting
40 90 73 81 35

Marks after sorting
35 40 73 81 90

RESULT:

Thus the program for sorting using pass by reference was executed successfully.

EX.NO:14**EMPLOYEE PAYROLL USING STRUCTURE****AIM:**

To write a c program to generate employee payroll using structures.

ALGORITHM:

Step-1 Start the program

Step-2 Create a Structure named Employee containing records Id,name, Basic Salary, Net Salary, HRA, DA and Tax.

Step-3 Get the employee Id and retrieve his details.

Step-4 Using the basic salary, DA, HRA, Tax calculate tax and net salary

Step-5 Display the details each employee record containing name, number, basic salary, HRA, DA, Net Salary and Tax

Step-6 Stop.

PROGRAM:

```
#include<stdio.h>
#include<dos.h>
struct employee
{
    int NO;
    char NAME[10];
    int DESIGN_CODE;
    int DAYS_WORKED;
}EMP[12]={
    {1,"GANESH",1,25},
    {2,"MAHESH",1,30},
    {3,"SURESH",2,28},
    {4,"KALPESH",2,26},
    {5,"RAHUL",2,24},
    {6,"SUBBU",2,25},
    {7,"RAKESH",2,23},
    {8,"ATUL",2,22},
    {9,"DHARMESH",3,26},
    {10,"AJAY",3,26},
    {11,"ABDUL",3,27},
    {12,"RASHMI",4,29}
};
void main()
{

    int EMPNO;
    void gen_payslip(int);
    clrscr();

    printf("ENTER THE EMPLOYEE NO TO GENERATE PAYSALIP : ");
    scanf("%d",&EMPNO);
    if(EMPNO>0 && EMPNO<13)
        gen_payslip(EMPNO);
```


Employee name : Robin
Employee Id : 100
Employee Basic salary : 30000.000000
HRA : 3000.000000
DA : 10,500.000000
Tax : 4500.000000
Gross salary : 39000.000000

RESULT:

Thus the c program to generate employee payroll using structures was performed and executed successfully.

EX.NO:15**INTERNAL MARKS****AIM:**

Write a C program to display the student internal marks using structures.

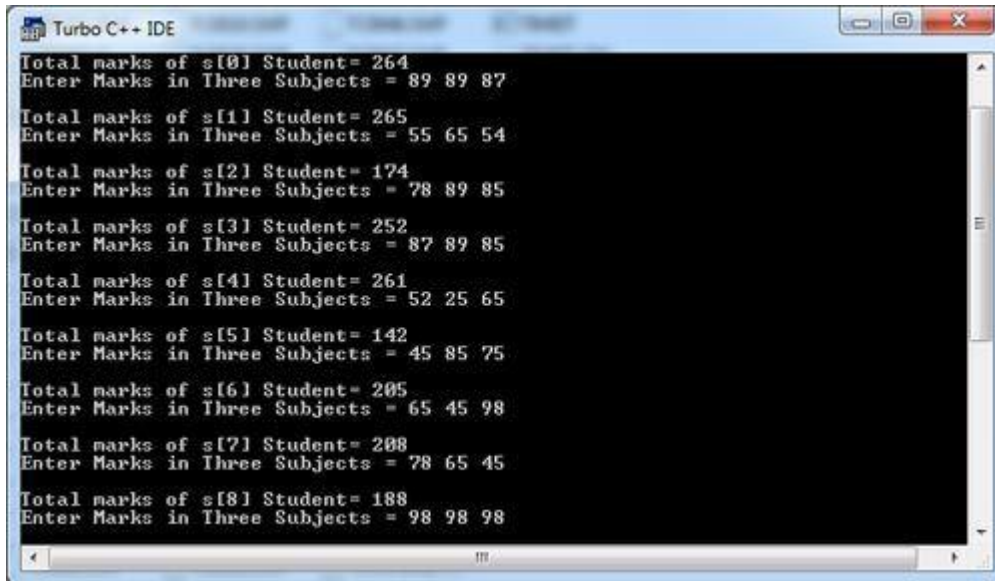
ALGORITHM:

- Step-1: Read the values of internal marks
- Step-2: Calculate internal marks of each student
- Step-3: Print the marks of each student
- Step-4: Stop

PROGRAM:

```
#include<stdio.h>
struct student
{
    int sub1;
    int sub2;
    int sub3;
};
void main()
{
    struct student s[10];
    int i,total=0;
    clrscr();
    for(i=0;i<=2;i++)
    {
        printf("\nEnter Marks in Three Subjects = ");
        scanf("%d%d%d",& s[i].sub1,&s[i].sub2,&s[i].sub3);
        total=s[i].sub1+s[i].sub2+s[i].sub3;
        printf("\nTotal marks of s[%d] Student= %d",i,total);
    }
    getch();
}
```

OUTPUT:



```
Turbo C++ IDE
Total marks of s[0] Student= 264
Enter Marks in Three Subjects = 89 89 87
Total marks of s[1] Student= 265
Enter Marks in Three Subjects = 55 65 54
Total marks of s[2] Student= 174
Enter Marks in Three Subjects = 78 89 85
Total marks of s[3] Student= 252
Enter Marks in Three Subjects = 87 89 85
Total marks of s[4] Student= 261
Enter Marks in Three Subjects = 52 25 65
Total marks of s[5] Student= 142
Enter Marks in Three Subjects = 45 85 75
Total marks of s[6] Student= 205
Enter Marks in Three Subjects = 65 45 98
Total marks of s[7] Student= 208
Enter Marks in Three Subjects = 78 65 45
Total marks of s[8] Student= 188
Enter Marks in Three Subjects = 98 98 98
```

RESULT:

Thus the program for student internal marks was executed successfully.

EX.NO:16**TELEPHONE DIRECTORY****AIM:**

Write a C program for telephone directory to perform insert, delete and append operations.

ALGORITHM:

Step-1: Read the record number,name and age of a person

Step-2: Insert the biodata in to the file

Step-3: Display the file

Step-4:Stop

PROGRAM:

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

struct biodata{
    int recno,age;
    char name[20],sex;
    float salary;
};

void main(){
    void addData(void);
    void delData(void);
    void showAll(void);
    void showRecord(void);
    void alterData(void);

    char choice;
    clrscr();

    while(1){
        clrscr();
        textcolor(BLACK);
        cprintf("    B I O - D A T A\r\n");
        printf("\n\n*****CHOOSE YOUR CHOICE*****\n");
        printf("1) ADD DATA\n");
        printf("2) DELETE DATA\n");
        printf("3) SHOW ALL\n");
        printf("4) SHOW RECORD\n");
        printf("5) ALTER DATA\n");
        printf("6) Exit \n");
        printf("Enter your choice : ");
        fflush(stdin);
        choice = getche();
        switch(choice){
            case '1' : //call add data
                addData();
                break;
            case '2' : //call delete databreak;
```

```

    case'3' : //call show all data
        showAll();
        break;
    case'4' : //call show record
        showRecord();
        break;
    case'5' : //call alter databreak;
    case'6' :
    case 27 : clrscr();
        gotoxy(25,10);
        _setcursortype(_NOCURSOR);
        textcolor(LIGHTMAGENTA);
        cprintf("THANKS FOR USING THIS SOFTWARE");
        getch();
        exit(1);
    }
}

//Adding Record to Filevoid addData(){
    FILE *fp;
    struct biodata obj;
    fp = fopen("biodata.txt","a+t");
    clrscr();
    printf("\n*****ADDING DATA*****\n");
    printf("\nEnter Record No : ");
    scanf("%d",&obj.recno);
    printf("Enter Name : ");
    fflush(stdin);
    scanf("%s",obj.name);
    printf("Enter age : ");
    scanf("%d",&obj.age);
    fflush(stdin);
    printf("Enter Sex : ");
    scanf("%c",&obj.sex);
    printf("Enter Salary : ");
    scanf("%f",&obj.salary);
    fscanf(stdin,"%d %s %d %c %f",&obj.recno,obj.name,&obj.age,&obj.sex,&obj.salary);
    fprintf(fp,"%d %s %d %c %f",obj.recno,obj.name,obj.age,obj.sex,obj.salary);
    fclose(fp);
}

void showRecord(){
    FILE *fp;
    struct biodata obj;
    int rec;
    long pos;
    fp = fopen("biodata.txt","r");
    clrscr();
    printf("\n*****SHOWING SPECIFIC RECORD*****\n");
    printf("\nEnter Record No : ");

```

```

scanf("%d",&rec);
pos = rec * sizeof(obj);
fseek(fp,pos,SEEK_SET);
if(feof(fp)==0)
printf("\n\nNO DATA FOUND\n");
else{
fscanf(fp,"%d %s %d %c %f",&obj.recno,obj.name,&obj.age,&obj.sex,&obj.salary);
printf("\n\n\tRecord No : %d\n",obj.recno);
printf("\tName : %s\n",obj.name);
printf("\tAge : %d\n",obj.age);
printf("\tSex : %c\n",obj.sex);
printf("\tSalary : %f\n",obj.salary);
}
getch();
fclose(fp);
}

void showAll(){
FILE *fp;
struct biodata obj;
int totrec,i;
fp = fopen("biodata.txt","r");
clrscr();
fseek(fp,0,SEEK_END);
totrec=ftell(fp)/sizeof(obj);
printf("\n*****SHOWING ALL RECORD*****\n");
printf("\n\nRecord_No\tName\tAge\tSex\tSalary\n\n");
printf("\n\n%d\n",totrec);
for(i=1;i<=totrec;i++){
fscanf(fp,"%d %s %d %c %f",&obj.recno,obj.name,&obj.age,&obj.sex,&obj.salary);
fprintf(stdout,"%-15d %-15s %-8d %-2c %10.2f\n",obj.recno,obj.name,obj.age,obj.sex,obj.salary);
}
getch();
fclose(fp);
}

```

OUTPUT:

```

B I O - D A T A
CHOOSE YOUR CHOICE
(1) ADD DATA\n");
(2) DELETE DATA\n");
(3) SHOW ALL\n");
(4) SHOW RECORD\n");
(5) Exit
1
("\n*****ADDING DATA*****\n");
("\nEnter Record No : ");
234
("Enter Name : ");
Raja
("Enter age : ");

```

30

```
35  
("Enter Sex : ");  
F  
("Enter Salary : ");  
35000
```

RESULT:

Thus the program for telephone details was executed successfully.

EX.NO:17**BANK ACCOUNT DETAILS****AIM:**

Write a C program to count the number of account holders whose balance is less than the minimum balance using sequential access file.

ALGORITHM:

Step-1: Read the account holder's name

Step-2: Check the account holder's balance

Step-4: Whether the balance is low means display the account holders name.

Step-5.Stop

PROGRAM:

* Program in C to show the bank operation using structure with array and Function. *\

```
# include < stdio.h >
# include < conio.h >
void creation();
void deposit();
void withdraw();
void lowbal();
int a = 0, i = 1001;
struct bank
{
int no;
char name[20];
float bal;
float dep;
} s[100];

int main()
{
int ch;
do
{
printf(" \n***** ");
printf(" \nBANKING ");
printf(" \n***** ");
printf(" \n1. Create New Account ");
printf(" \n2. Cash Deposit ");
printf(" \n3. Cash Withdraw ");
printf(" \n4. Low Balance Enquiry ");
printf(" \n5. Exit ");
printf(" \nEnter your choice : ");
scanf("%d",& ch);
switch ( ch)
{
case 1 : creation();
break;
case 2 : deposit();
```

```

break ;
case 3 : withdraw( ) ;
break ;
case 4 : lowbal( ) ;
break ;
case 5 : ;
break ;
defalut : printf(" Choice a Valid option !! ") ;
break ;
getch( ) ;
}
} while( ch != 5 ) ;
}

void creation( )
{
printf(" \n***** ") ;
printf(" \nNEW ACCOUNT CREATION ") ;
printf(" \n***** ") ;
printf(" \nYour Account Number is :%d ",i) ;
s[a].no = i ;
printf(" \nEnter your Name : ") ;
scanf("%s ",& s[a].name) ;
printf(" \nYour Deposit is Minimum Rs.500") ;
s[a].dep=500 ;a++ ;
i++ ;
getch( ) ;
}

void deposit( )
{
int no, b = 0, m = 0 ;
int aa ;
printf(" \n***** ") ;
printf(" \nCASH DEPOSIT ") ;
printf(" \n***** ") ;
printf(" \nEnter your Account Number : ") ;
scanf("%d",& no) ;
for ( b = 0 ; b < i ; b++)
{
if ( s[b].no == no)
m = b ;
}
if ( s[m].no == no)
{
printf("\n Account Number : %d ",s[m].no) ;
printf("\n Name : %s ",s[m].name) ;
printf("\n Deposit : %f ",s[m].dep) ;
printf(" \nDeposited Amount : ") ;
scanf("%f",& &aa) ;
s[m].dep+=aa ;
printf("\nThe Balance in Account is : %f ",s[m].dep) ;
getch( ) ;
}
}

```



```

}
else
{
printf("\nACCOUNT NUMBER IS INVALID ");
getch();
}
}
void withdraw()
{
int no, b = 0, m = 0;
int aa;
printf(" \n***** ");
printf(" \nCASH WITHDRAW ");
printf(" \n***** ");
printf(" \nEnter your Account Number : ");
scanf("%d",& no);
for ( b = 0 ; b < i ; b++)
{
if ( s[b].no == no)
m = b;
}
if ( s[m].no == no)
{
printf("\n Account Number : %d ",s[m].no);
printf("\n Name : %s ",s[m].name);
printf("\n Deposit : %f ",s[m].dep);
printf(" \nWithdraw Amount : ");
scanf("%f",& aa);
if ( s[m].dep < aa+500)
{
printf("\nCANNOT WITHDRAW YOUR ACCOUNT HAS MINIMUM BALANCE ");
getch();
}
else
{
s[m].dep-=aa;
printf("\nThe Balance Amount in Account is : %f ",s[m].dep);
}
}
else
{
printf("\nACCOUNT NUMBER IS INVALID ");
getch();
}
getch();
}
void lowbal()
{
int no, b = 0, m = 0;
int aa;
printf(" \n***** ");
printf(" \nFOLLOWING ACCOUNT HOLDER'S BALANCE IS LESS THAN 1000 ");

```

```

printf("\n*****");
for ( b = 0 ; b < a ; b++)
{
if ( s[b].dep < 1000))
{
printf("\n\n Account Number : %d ",s[b].no) ;
printf("\n Name : %s ",s[b].name) ;
}
}
getch() ;
}

```

OUTPUT:

```

Turbo C++ IDE
ENTER THE EMPLOYEE NO TO GENERATE PAYSALIP : 1

                SHREE KRISHNA CHEMISTS AND DRUGGIST
                SALARY MONTH 9 2013

EMP.NO.: 1           EMP.NAME: GANESH           DESIGNATION: CLERK
DAYS WORKED: 25     PAY RATE: 400           GEN.DATE:15/9/2013

-----
EARNINGS           AMOUNT(RS.)           DEDUCTIONS           AMOUNT(RS.)
-----
BASIC PAY           10000                P.F.                 1000
                                                  PROF.TAX             200
-----
GROSS EARN.         10000                TOTAL DEDUCT.        1200
                    NET PAY               8800
-----

```

RESULT:

Thus the program for account holder's minimum balance was checked and executed successfully.

EX.NO:18**RAILWAY RESERVATION SYSTEM (MINI PROJECT)****AIM:**

Write a C program for railway reservation system with availability checking, cancellation, prepare chart.

ALGORITHM:

Step-1: Read train number
 Step-2: Read total number of seats in first class
 Step-3: Read total number of seats in second class
 Step-4: Read total number of sets in third class
 Step -5: Check if it is any vacancy is there
 Step-6: Reverse the ticket
 Step-7 : Stop

PROGRAM:

```
#include<iostream.h>
#include<conio.h>
class Train { int number; //train number
int seats_1;//total seats in first class
int seats_2;// total seats in second class
int seats_3;//total seats in third class
public:
Train(int i,int j, int k, int l)
{ number = i; seats_1 = j;
seats_2=k ; seats_3 = l;
}
int getnum(void)
{ return number;}
int getseats_1(void)
{ return seats_1;}
int getseats_2(void)
{ return seats_2;}
int getseats_3(void)
{ return seats_3;}
};
class Reservation :public Train
{ int bkd_1;// seats reserved in 1st class
int bkd_2;//seats reserved in 2nd class
int bkd_3;// seats reserved in 3rd class
public:
int i,j,k,l;
Reservation (int i,int j,int k,int l):Train(i,j,k,l)
{ bkd_1=bkd_2=bkd_3=0;}
void book(char type,int num);
void cancel(char type,int num);
void disp_status(void);
};
void Reservation::book(char type,int num)
{ switch(type)
```

```

{ case'1': bkd_1+=num; //add num to bkd_1
break;
case'2': bkd_2+=num;//add num to bkd_2
break;
case'3': bkd_3+=num;//add num to bkd_3
break;
default: cout<<"wrong class\n";
}
}

```

```

void Reservation :: cancel(char type, int num)
{ switch(type)
{ case'1':bkd_1-=num;
break;
case'2':bkd_2-=num;
break;
default:cout<<"wrong class\n";
}
}

```

```

void Reservation::disp_status(void)
{ cout<<"\t\t Train number:"<<getnum()<<"\n";
cout<<"Class\tTotal seats\tReseved\tUn reserved\n";
int val;
val=getseats_1();
cout<<"1\t"<<val<<"\t"<<bkd_1<<"\t"<<val-bkd_1<<"\n";
val=getseats_2();
cout<<"2\t"<<val<<"\t"<<bkd_2<<"\t"<<val-bkd_2<<"\n";
val=getseats_3();
cout<<"3\t"<<val<<"\t"<<bkd_3<<"\t"<<val-bkd_3<<"\n";
}

```

```

void main()
{ clrscr();
int num;
cout<<"Enter Train Number:\n";
cin>>num;
cout<<"\n Enter total Number Of seats in 1st class\n";
int s1;
cin>>s1;
cout<<"\n Enter total number of seats in 2nd class\n";
int s2;
cin>>s2;
cout<<"\n Enter total number of seats in 3rd class\n";
int s3;
cin>>s3;
Reservation Tr(num,s1,s2,s3);
char cl_type;
int choice,seats;
do
{ cout<<"\n Main Menu\n";
cout<<"1.Reservation\n";

```

```
cout<<"2.Cancellation\n";
cout<<"3.Display Status\n";
cout<<"4.Exit\n";
cout<<"Enter your choice:";
cin>>choice;
cout<<"\n";
switch(choice)
{ case 1: cout<<"which class?(1/2/3):";
cin>>cl_type;
cout<<"\n How many seats?";
cin>>seats;
cout<<"\n";
Tr.book(cl_type,seats);
break;
case 2:cout<<"Which class?(1/2/3):";
cin>>cl_type;
cout<<"\n How many seats?";
cin>>seats;
cout<<"\n";
Tr.cancel(cl_type,seats);
break;
case 3:Tr.disp_status();
break;
case 4:break;
default : cout<<"Wrong choice!!\n";
}; //end of switch
} while(choice>=1&&choice<=3);
getch();
}
```

OUTPUT:

```
*****
BANKING
*****
1-Create New Account
2-Cash Deposit
3-Cash Withdraw
4-Low Balance Enquiry
5-Exit
Enter your choice : 1

*****
NEW ACCOUNT CREATION
*****
Your Account Number is :1001
Enter your Name:JIYA

Your Deposit is Minimum Rs.500
*****
BANKING
*****
1-Create New Account
2-Cash Deposit
3-Cash Withdraw
4-Low Balance Enquiry
5-Exit
Enter your choice : 2

*****
CASH DEPOSIT
*****
Enter your Account Number : 1001

Account Number : 1001
Name : JIYA
Deposit : 500.000000
Deposited Amount : 1000

The Balance in Account is :1500.000000
```

RESULT:

Thus the program for railway reservation system was executed successfully.