Extra Lab Exercise

1.Operator:-

→**Lab exercise 1**: Simple Calculator

- Write a C program that acts as a simple calculator. The program should take two numbers and an operator as input from the user and perform the respective operation (addition, subtraction, multiplication, division, or modulus) using operators.
- **Challenge**: Extend the program to handle invalid operator inputs

Ans.

```
#include<stdio.h>

int main(){

        int num1,num2;

        up:

        printf("\n enter the value of num1=");

        scanf("%d",&num1);

        printf("\n enter the value of num2=");

        scanf("%d",&num2);


        char ch='%',choice;

        printf("\n Press '+' for Addition");
```

```
    printf("\n Press '-' for Subtarction");
    printf("\n Press '*' for Multiplication");
    printf("\n Press '/' for Division");
    printf("\n Press '%c' for Remaindor",ch);
    float ans=(float)num1/(float)num2;

    printf("\n Enter your Choice...:");
    scanf(" %c",&choice);

    switch(choice){
        case '+':
            printf("\n Addition of %d and %d is =%d",num1,num2,num1+num2);
            break;
        case '-':
            printf("\n Subtraction of %d and %d is =%d",num1,num2,num1-num2);
            break;
        case '*':
            printf("\n Multiplication of %d and %d is =%d",num1,num2,num1*num2);
            break;
        case '/':
            printf("\n Division of %d and %d is
=%.2f",num1,num2,ans);
            break;
        case '%':
            printf("\n Remaindor of %d and %d is
=%d",num1,num2,num1%num2);
            break;
        default:
            printf("\n Invalid input");
            printf("\n Try again!");
            goto up;
    }
    printf("\n Do you Want to Perform Another Calculation?
Y/N...:");
    scanf(" %c",&choice);
    if(choice=='y'||choice=='Y'){
        goto up;
    }



    return 0;
```

```
}
```

→ **Lab exercise 2**: Check Number Properties.

→Write a C program that takes an integer from the user and checks the following using different operators:

1. Whether the number is even or odd.
2. Whether the number is positive, negative, or zero.
3. Whether the number is a multiple of both 3 and 5.

Ans.

```
/*Write a C program that takes an integer from the user and
checks the following using

different operators:

o Whether the number is even or odd.

o Whether the number is positive, negative, orzero.

o Whether the number is a multiple of both 3 and 5.

*/

#include<stdio.h>

int main(){

	int num;

	printf("\n enter the number=");

	scanf("%d",&num);
```

```
        int c;

        c=(num%2==0)?printf("\n %d is even
number",num):printf("\n %d is  a odd number",num);


        int d;

        d=(num>0)?printf("\n %d is positive
number",num):(num<0)?printf("\n %d is negative
number",num):printf("\n number is neutral");


        int f=(num%3==0&&num%5==0)?printf("\n %d is multiple
of both 3 and 5",num):printf("\n %d is not a multiple of both 3
and 5",num);

        return 0;

}
```

2.Conditinal statement:-

→**Lab exercise 1**:Grade Calculator

• Write a C program that takes the marks of a student as input and displays the corresponding grade based on the following conditions:

i. Marks > 90: Grade A
ii. Marks > 75 and <= 90: Grade B
iii. Marks > 50 and <= 75: Grade C
iv. Marks <= 50: Grade D •

Use if-else or switch statements for the decision-making process.

Ans.

```
#include<stdio.h>

Int main(){

Int marks;

Char grade;

Printf("\n enter the marks of student=");

Scanf("%d",&marks);

If(marks>=90){

Grade='a';}

 Else if(marks>70&&marks<90){
  Grade=’b’;
}
Else if(marks>50&&marks<70){
  Grade=’c’;
}
Else{
Grade=’d’;
}
Printf(“\n grade of student=%c”,grade);

Return 0;}
```

→**LAB EXERCISE 2**: Number Comparison

Write a C program that takes three numbers from the user and determines:

1 .The largest number.

2. The smallest number.

Challenge: Solve the problem using both if-else and switch-case statements.

Ans.

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

```
//using if-else
int n1,n2,n3;
printf("\n enter the num1=");
scanf("%d",&n1);
printf("\n enter the num2=");
scanf("%d",&n2);
printf("\n enter the num3=");
scanf("%d",&n3);
if(n1>n2&&n1>n3){
        printf("\n %d is a largest number",n1);

}
else if(n2>n1&&n2>n3){

        printf("\n %d is a largest number",n2);
}
else{
        printf("\n %d is a largest number",n3);
}
if(n1<n2&&n1<n3){
        printf("\n %d is a smallest number",n1);
}
else if(n2<n1&&n2<n3){

        printf("\n %d is a smallest number",n2);
}
else{

        printf("\n %d is a smallest number",n3);
}



//using switch statement
//int large,small;
switch(n1>n2&&n1>n3){
        case 1:

            printf("\n %d is a largest number",n1);
                break;
        case 0:
                switch(n2>n1&&n2>n3){
                    case 1:

                  printf("\n %d is a largest number",n2);
                        break;
                    case 0:

                  printf("\n %d is a largest number",n3);
                        break;
                }
        }
        switch(n1<n2&&n1<n3){
            case 1:

              printf("\n %d is a smallest number",n1);
                break;
            case 0:
                switch(n2<n1&&n2<n3){
                    case 1:

                  printf("\n %d is a smallest number",n2);
                        break;
                    case 0:

                  printf("\n %d is a smallest number",n3);
                        break;
                }
        }
        return 0;
    }
    --------------------------------------------------------------------------------
```

3.Loops:-

→ **Lab Exercise 1:** Prime Number Check

- Write a C program that checks whether a given number is a prime number or not using a for loop.
- Challenge: Modify the program to print all prime numbers between 1 and a given number.

Ans.

```
#include<stdio.h>
int main(){
    //prime number check
    int num,i,j,flag=1;
    printf("\n enter the number=");
    scanf("%d",&num);

    for(i=2;i<num;i++){
        if(num%i==0){
            flag=0;
          break;
        }

    }
    if(flag==1){
            printf("\n %d is a prime number",num);
    }
    else{
            printf("\n %d is not a prime number",num);
    }

printf("\nPrime numbers between 1 and %d are:\n", num);

for (i = 2; i <= num; i++) {
    flag=1;
    for (j = 2; j <= i / 2; j++) {
        if (i % j == 0) {
            flag= 0;
            break;
       }
    }

    if (flag==1) {
       printf("%d ", i);
    }
}
```

```
	return 0;

}
```

→**LAB EXERCISE 2:** Multiplication Table

• Write a C program that takes an integer input from the user and prints its multiplication table using a for loop.

• Challenge: Allow the user to input the range of the multiplication table (e.g., from 1 to N).

Ans.

```
#include<stdio.h>
int main(){
	int num,range,i;
	printf("\n enter the number for print its multiplication table=");
	scanf("%d",&num);

	printf("\n enter the range for multiplication=");
	scanf("%d",&range);
	printf("\n multiplication table of %d from 1 to 
%d=\n",num,range);
	for(i=1;i<=range;i++){
	   printf("\n %d*%d=%d",num,i,num*i);
	}
	return 0;
}
```

→**LAB EXERCISE 3**: Sum of Digits

- Write a C program that takes an integer from the user and calculates the sum of its digits using a while loop.
- Challenge: Extend the program to reverse the digits of the number.

Ans.

```
#include<stdio.h>
int main(){
	int num,sum=0,rem,rev=0;
	printf("\n enter the number=");
	scanf("%d",&num);
	printf("\n number you entered=%d",num);
	while(num!=0){
		rem=num%10;
```

```
        sum=sum+rem;

        rev=(rev*10)+rem;

        num=num/10;

    }

    printf("\n sum of all its digit=%d",sum);


    printf("\n reversed digit of number=%d",rev);

    return 0;

}
```

4.Array:-

→ **LAB EXERCISE 1**: Maximum and Minimum in Array

- Write a C program that accepts 10 integers from the user and stores them in an array. The program should then find and print the maximum and minimum values in the array.

- Challenge: Extend the program to sort the array in ascending order.

Ans.

```
#include<stdio.h>

int main(){

        int a[10],i;

        for(i=0;i<10;i++){

                printf("\n enter the element in a[%d]",i);

                scanf("%d",&a[i]);

        }

        int max=a[0];

    int min=a[0];

    for(i=0;i<10;i++){

        if(a[i]>max){

                max=a[i];

                }

        }
```

```
for(i=0;i<10;i++){
        if(a[i]<min){
                min=a[i];
        }
}
printf("\n maximum value is =%d",max);
printf("\n miniimum value is =%d",min);
printf("\n original array=\n");
for(i=0;i<10;i++){
        printf("%d ",a[i]);
}
int j,temp;
for(i=0;i<10;i++){
        for(j=i+1;j<10;j++){
                if(a[i]>a[j]){
                        temp=a[i];
                        a[i]=a[j];
                        a[j]=temp;
                }
        }
}
```

```
    printf("\n sorted array=\n");
    for(i=0;i<10;i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

→**LAB EXERCISE 2:** Matrix Addition

- Write a C program that accepts two 2x2 matrices from the user and adds them. Display the resultant matrix.
- Challenge: Extend the program to work with 3x3 matrices and matrix multiplication.

Ans.

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

    int a[10][10],b[10][10],ans[10][10];
    int size,i,j,k;
    printf("\n enter the size of matrix for Addition=");
    scanf("%d",&size);
    for(i=0;i<size;i++){
        for(j=0;j<size;j++){
                printf("\n enter the element in a[%d][%d]=",i,j);
                scanf("%d",&a[i][j]);
        }
}
for(i=0;i<size;i++){
        for(j=0;j<size;j++){
                printf("\n enter the element in b[%d][%d]=",i,j);
                scanf("%d",&b[i][j]);
        }
}
for(i=0;i<size;i++){
        for(j=0;j<size;j++){
                ans[i][j]=a[i][j]+b[i][j];
        }
}

printf("\n array a=\n");
for(i=0;i<size;i++){
        for(j=0;j<size;j++){
                printf("%d ",a[i][j]);
        }
```

```
    printf("\n");
}
printf("\n array b=\n");
for(i=0;i<size;i++){
    for(j=0;j<size;j++){
        printf("%d ",b[i][j]);
    }
    printf("\n");
}
printf("\n matrix addition=\n");
for(i=0;i<size;i++){
    for(j=0;j<size;j++){
        printf("%d ",ans[i][j]);
    }
    printf("\n");
}
printf("\n aaddition completed of 2*2 matrix.");
printf("\n now try 3*3 matrix Multiplication");



int c[10][10],d[10][10],mul[10][10];
int size1;
printf("\n enter the size of matrix for multiplication=");
scanf("%d",&size1);
for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
                printf("\n enter the element in c[%d][%d]=",i,j);
                scanf("%d",&c[i][j]);
        }
}
for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
                printf("\n enter the element in d[%d][%d]=",i,j);
                scanf("%d",&d[i][j]);
        }
}
for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
                mul[i][j]=0;
                for(k=0;k<size1;k++){
                        mul[i][j]=mul[i][j]+(c[i][k]*d[k][j]);
                }
        }
    }

    printf("\n array c=\n");
    for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
            printf("%d ",c[i][j]);
        }
        printf("\n");
    }
    printf("\n array d=\n");
    for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
            printf("%d ",d[i][j]);
        }
        printf("\n");
    }
    printf("\n matrix multiplication=\n");
    for(i=0;i<size1;i++){
        for(j=0;j<size1;j++){
            printf("%d ",mul[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

→**LAB EXERCISE 3:** Sum of Array Elements

Write a C program that takes N numbers from the user and stores them in an array. The

program should then calculate and display the sum of all array elements.

Challenge: Modify the program to also find the average of the numbers.

```
#include<stdio.h>
int main(){
    int a[100],size,i,sum=0;
    printf("\n enter the size of array=");
    scanf("%d",&size);
    for(i=0;i<size;i++){
        printf("\n enter the element in a[%d]=",i);
        scanf("%d",&a[i]);
    }
    printf("\n original array=");
    for(i=0;i<size;i++){
        printf("%d ",a[i]);
    }

    for(i=0;i<size;i++){
        sum=sum+a[i];
    }
    printf("\n sum of all element=%d",sum);

    float avg=(float)sum/(float)size;
    printf("\n average of the numbers=%.2f",avg);
    return 0;
}
```

→5. Functions:-

→**LAB EXERCISE 1**: Fibonacci Sequence

? Write a C program that generates the Fibonacci sequence up to N terms using a recursive

function.

? Challenge: Modify the program to calculate the Nth Fibonacci number using both iterative

and recursive methods. Compare their efficiency.*/

```
#include<stdio.h>
void fibo_recu(int terms,int n1,int n2);
int fibo_rec_nth(int terms,int n1,int n2);
void fibo_ite_nth(int terms,int n1,int n2);
int main(){
	int terms;
	int n1=0,n2=1;
	printf("\n enter the terms for fibonacci series=");
	scanf("%d",&terms);
	printf("\n fibonacci series=%d %d ",n1,n2);


	fibo_recu(terms-2,n1,n2);
	int nth=fibo_rec_nth(terms-1,n1,n2);


	printf("\n fibonacci series of nth term= %d ",nth);


	printf("\n fibonacci series of nth term=  ");


	fibo_ite_nth(terms,n1,n2);
	return 0;
}
void fibo_recu(int terms,int n1,int n2){
	if(terms<=0){
		return;
	}
	int n3;
		n3=n1+n2;
		n1=n2;
		n2=n3;
		printf("%d ",n3);
	  fibo_recu(terms-1,n1,n2);
```

```
}
int fibo_rec_nth(int terms,int n1,int n2){
        int n3;
        if(terms<=0){
        return;
                //printf("%d ",n3);
        }

        n3=n1+n2;

        n1=n2;
        n2=n3;

        fibo_rec_nth(terms-1,n1,n2);

}

void fibo_ite_nth(int terms,int n1,int n2){
        int n3,i;
        if(terms<=0){
                return;
```

```
    }
    for(i=0;i<terms-1;i++){
        n3=n1+n2;
        n1=n2;
        n2=n3;

    }
    printf("%d ",n3);
}
```

→**LAB EXERCISE 2:** Factorial Calculation

? Write a C program that calculates the factorial of a given number using a function.

? Challenge: Implement both an iterative and a recursive version of the factorial function and

compare their performance for large numbers.*/

```
#include<stdio.h>
void fact(int num);
void fact_rec(int num,int fact1);
int main()
{
        int num,fact1=1;
        printf("\n enter the num=");
        scanf("%d",&num);
        fact(num);
    fact_rec(num,fact1);
        return 0;
}
void fact(int num){
        int fact=1,i;
        for(i=1;i<=num;i++){
                fact=fact*i;
        }
        printf("\n factorial of %d is =%d",num,fact);
}
void fact_rec(int num,int fact1){
        //int i=1;
        if(num==0||num==1){
                printf("\n factorial is %d",fact1);
```

```
            return;
        }
        fact1=fact1*num;
        fact_rec(num-1,fact1);



}
```

→**LAB EXERCISE 3:** Palindrome Check

? Write a C program that takes a number as input and checks whether it is a palindrome using

a function.

? Challenge: Modify the program to check if a given string is a palindrome.*/

```
#include<stdio.h>
#include<string.h>
void n_pali(int num);
void s_pali(char str[]);
int main(){
        int num;
        printf("\n enter the num=");
        scanf("%d",&num);
        n_pali(num);
        char str[100];
        printf("\n enter any string=");
        scanf("%s",str);
        s_pali(str);
        return 0;
}

void n_pali(int num){
        int rem,rev=0,i;
        int temp=num;
        while(num!=0){
                rem=num%10;
                rev=(rev*10)+rem;
                num=num/10;
        }
        if(temp==rev){
                printf("\n %d is a palindrome number",temp);
        }
        else{
```

```
		printf("\n %d is not a palindrome number",temp);
	}
}

void s_pali(char str[]){
	int len=0,i;
	char str1[100];
	strcpy(str1,str);
	strrev(str);
	int res=strcmp(str1,str);
	if(res==0){
		printf(" %s is a palindrome string",str1);
	}
	else{
		printf(" %s is not a palindrome strring",str1);
	}
}
```

→6. Strings

→LAB EXERCISE 1: String Reversal

- Write a C program that takes a string as input and reverses it using a function.
- Challenge: Write the program without using built-in string handling functions.

Ans.

```
#include<stdio.h>
void reverse(char str[]);
int main(){
    char str[50];
    printf("\n enter Any string=");
    gets(str);
    printf("\n original string=%s",str);
    reverse(str);
    return 0;
}
void reverse(char str[]){
    int i,len=0;
    for(i=0;str[i]!='\0';i++){
        len++;
    }
```

```
	printf("\n Reversed string=");
	for(i=len-1;i>=0;i--){
		printf("%c",str[i]);
	}
}
```

→**LAB EXERCISE 2:** Count Vowels and Consonants

- Write a C program that takes a string from the user and counts the number of vowels and consonants in the string.
- Challenge: Extend the program to also count digits and special characters.

Ans.

```
#include<stdio.h>
#include<ctype.h>
int main(){
	char str[50];
	int vowel=0,consonant=0,digit=0,sc=0,i;
	printf("\n enter the string=");
	gets(str);
	for(i=0;str[i]!='\0';i++){


		if(isalpha(str[i])){
       if(str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u'||str[i]=='A'||str[i]=='E'||str[i]=='I'||str[i]=='O'||str[i]=='U'){
              vowel++;
       }


          else{
          consonant++;
             }



       }
       else if(isdigit(str[i])){
            digit++;
         }
         else if(str[i]!=' '&& str[i]!='\n'){
            sc++;
         }
   }
        printf("\n number of vowel in string=%d",vowel);
     printf("\n cconsonant in string=%d",consonant);
```

```
    //printf("\n cconsonant in string=%d",consonant);
    printf("\n digit in string=%d",digit);
    printf("\n special character in string=%d",sc);



        return 0;
}
```

→**LAB EXERCISE 3:** Word Count

• Write a C program that counts the number of words in a sentence entered by the user.

• Challenge: Modify the program to find the longest word in the sentence.

Ans.

```
#include<stdio.h>
int main(){
        char str[1000];
        int word=0,i;
        printf("\n enter the sentence =");
        fgets(str,sizeof(str),stdin);
        for(i=0;str[i]!='\0';i++){
                if(str[i]==' '||str[i]=='\n'){
```

```
                word++;


                continue;
            }


        }
        printf("\n words=%d",word);
        return 0;
}
```