

C PROGRAMMING LABORATORY

Semester I/II

Course Code : 18CPL17/27

Teaching Hours/week (L:T:P) : 0:0:2

CIE Marks :40

SEE Marks :60

Exam Hours:03

Credits : 01

Laboratory Programs:

1. Familiarization with computer hardware and programming environment, concept of naming the program files, storing, compilation, execution and debugging, taking any simple C- code.

PART A

2. Develop a program to solve simple computational problems using arithmetic expressions and use of each operator leading to simulation of a commercial calculator. (No built-in math function)

3. Develop a program to compute the roots of a quadratic equation by accepting the coefficients. Print appropriate messages.

4. Develop a program to find the reverse of a positive integer and check for palindrome or not. .Display appropriate messages.

5. An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.

6. Introduce 1D Array manipulation and implement Binary search.

7. Implement using functions to check whether the given number is prime and display appropriate messages. (No built-in math function).

PART B

8. Develop a program to introduce 2D Array manipulation and implement Matrix multiplication and ensure the rules of multiplication are checked.

9. Develop a Program to compute Sin(x) using Taylor series approximation. Compare your result with the built- in Library function. Print both the results with appropriate messages.

10. Write functions to implement string operations such as compare, concatenate, string length. Convince the parameter passing techniques.

11. Develop a program to sort the given set of N numbers using Bubble sort.

12. Develop a program to find the square root of a given number N and execute for all possible inputs with appropriate messages. Note: Don't use library function sqrt(n).

13. Implement structures to read, write and compute average- marks and the students scoring above and below the average marks for a class of N students.

14. Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

15. Implement Recursive functions for Binary to Decimal Conversion.

Marks Distribution

a For questions having only one part —

b Procedure + Execution + Viva- Voce: $15+70+15=100$ Marks

Part A—Procedure + Execution + Viva= $4+21+5=30$ Marks

Part B—Procedure + Execution + Viva= $10+49+ 11 =70$ Marks

PART A

2. Develop a program to solve simple computational problems using arithmetic expressions and use of each operator leading to simulation of a commercial calculator. (No built-in math function)

```
#include<stdio.h>
#include<stdlib.h>
main()
{
float num1,num2,result;
char op;
printf("Enter the operator(+,-,*,./)\n");
scanf("%c",&op);
printf("Please enter two numbers:\n");
scanf("%f%f", &num1,&num2);
if(op=='+')
{
result=num1+num2;
printf("result is %0.2f\n",result);
}
else if(op=='-')
{
result=num1-num2;
printf("result is %0.2f\n",result);
}
else if(op=='*')
{
result=num1*num2;
printf("result is %0.2f\n",result);
}
else
{
{
if(num2==0)
{
printf("please enter non zero no. For num2:\n");
exit(0);
}
else
result=num1/num2;
printf("result is %0.2f\n",result);
}
}
}
```

OUTPUT:

Enter the operator (+, -, *, /)

+

Please enter two numbers : 4 2

6.00

Enter the operator (+, -, *, /)

-

Please enter two numbers : 4 2

2.00

Enter the operator (+, -, *, /)

*

Please enter two numbers : 4 2

8.00

Enter the operator (+, -, *, /)

/

Please enter two numbers : 4 2

2.00

3. Develop a program to compute the roots of a quadratic equation by accepting the coefficients. Print appropriate messages.

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a,b,c,d;
    float root1,root2,real,img;
    printf("\nEnter the values of a b and c:\t");
    scanf("%f%f%f",&a,&b,&c);
    if((a==0)&&(b==0))
    {
        printf("\nInvalid inputs");
    }
    else
    {
        d=(b*b)-(4*a*c);
        if(d==0)
        {
            printf("\nRoots are real and equal::");
            root1=root2=-b/(2*a);
            printf("\nRoot1=%.3f\nRoot2=%.3f",root1,root2);
        }
        else if(d>0)
        {
            printf("\nRoots are real and distinct::");
            root1=(-b+sqrt(d))/(2*a);
            root2=(-b-sqrt(d))/(2*a);
            printf("\nRoot1=%.3f\nRoot2=%.3f",root1,root2);
        }

        else
        {
            printf("\nRoots are real and imaginary::");
            real=-b/(2*a);
            img=(sqrt(fabs(d)))/(2*a);
            printf("\nRoot1=%.3f+i%.3f",real,img);
            printf("\nRoot2=%.3f-i%.3f",real,img);
        }
    }
}
```

OUTPUT:

1)Enter the values of a b and c: **1 6 9**

Roots are real and equal::

Root1 = **-3.000**

Root2 = **-3.000**

2)Enter the values of a b and c: **1 -5 3**

Roots are real and distinct::

Root1 = **4.303**

Root2 = **0.697**

3)Enter the values of a b and c: **1 4 7**

Roots are real and imaginary::

Root1 = **-2.000+i1.732**

Root2 = **-2.000-i1.732**

4. Develop a program to find the reverse of a positive integer and check for palindrome or not
Display appropriate messages.

```
#include<stdio.h>
main()
{
    int num, rev=0, rem, n;
    printf("\nEnter a number");
    scanf("%d", &num);
    n=num;
    while(num!=0)
    {
        rem=num%10;
        rev=rev*10+rem;
        num=num/10;
    }
    printf("\nReversed number is%d", rev);
    if(rev==n)
    {
        printf("\nGiven number is Palindrome");
    }
    else
    {
        printf("\nGiven number is not Palindrome ");
    }
}
```

OUTPUT:

- 1) Enter a number: **1221**
Reversed number is **1221**
Given number is Palindrome

- 2) Enter a number: **1234**
Reversed number is **4321**
Given number is not Palindrome

5. An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.

```
#include<stdio.h>
#include<string.h>
void main()
{
    char name[20];
    float unit,amt,charge;
    printf("Enter the name of the custmer");
    fgets(name,20,stdin);
    printf("Enter the total unit");
    scanf("%f", &unit);
    if(unit<=200)
    {
        amt=0.80*unit;
    }
    else if(unit<=300)
    {
        amt=160+0.90*(unit-200);
    }
    else
    {
        amt=250+1*(unit-300);
    }
    amt=amt+100;
    if(amt>400)
    {
        charge=amt*0.15;
    }
    else
    {
        charge=0;
    }
    amt=amt+charge;
    printf("Custmer name is %s and electric bill is %0.2f", name, amt);
}
```


OUTPUT:

1.Enter the name of the custmer - Asha

Enter the total unit – 100

Total bill - 180

2.Enter the name of the custmer - Rama

Enter the total unit –120

Total bill - 196.00

6. Introduce 1D Array manipulation and implement Binary search.

```
#include<stdio.h>

void main()
{
    int a[100],n,key,pos,i,mid;
    int first,last,found;
    printf("Enter No. of array elements\n");
    scanf("%d",&n);
    printf("Enter the Values\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    first=0;
    last=n-1;
    found=0;
    printf("Enter Key value to be search\n");
    scanf("%d",&key);
    while(first<=last)
    {
        mid=(first+last)/2;
        if (key==a[mid])
        {
            pos=mid+1;
            found=1;
            break;
        }
        else if(key<a[mid])
        {

```

```
last=mid-1;
}
else
{
first=mid+1;
}
}
if(found==1)
{
printf("key found at the %d position\n",pos);
}
else
{
printf("Key is found\n");
}
}
```

OUTPUT :

Enter No. of array elements : 5

Enter the Values : 7 5 3 9 4

Enter Key value to be search : 9

Key is found at the 4 position.

7. Implement using functions to check whether the given number is prime and display appropriate messages. (No built-in math function)

```
#include<stdio.h>

void main()
{
    int num,count;

    printf("Enter a number\n");

    scanf("%d",&num);

    count=prime(num);

    if(count>2)

        printf("%d is not a prime no.\n",num);

    else

        printf("%d is a prime no.\n",num);

}

int prime(num)

{

    int count=0;

    int i;

    for(i=1;i<=num;i++)

    {

        if(num%i==0)

        {

            count++;

        }

    }

    return count;

}
```

OUTPUT:

1. Enter a Number : **7**

7 is a Prime number.

2. Enter a Number : **8**

8 is not a Prime number.

8. Develop a program to introduce 2D Array manipulation and implement Matrix multiplication and ensure the rules of multiplication are checked.

```
#include<stdio.h>

void main()
{
    int a[10][10],b[10][10],i,j,p,q,m,n,c[10][10],k;

    printf("Enter the no of rows and columns of matrix a\n");

    scanf("%d%d",&m,&n);

    printf("Enter the no of rows and columns in matrix b\n");

    scanf("%d%d",&p,&q);

    printf("Enter the element of matrix a\n");

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    printf("Enter the elements of matrix b\n");

    for(i=0;i<p;i++)
    {
        for(j=0;j<q;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }

    printf("First matrix a\n");

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
```

```

    {
        printf("%d\t",a[i][j]);
    }
    printf("\n");
}
printf("Second matrix b\n");
for(i=0;i<p;i++)
{
    for(j=0;j<q;j++)
    {
        printf("%d\t",b[i][j]);
    }
    printf("\n");
}
if(n==p)
{
    for(i=0;i<m;i++)
    {
        for(j=0;j<q;j++)
        {
            c[i][j]=0;
            for(k=0;k<m;k++)
            {
                c[i][j]=c[i][j]+(a[i][k]*b[j][k]);
            }
        }
    }
}
printf("Resultant matrix is \n");
for(i=0;i<m;i++)

```

```

{
    for(j=0;j<q;j++)
    {
        printf("%d\t",c[i][j]);
    }
    printf("\n");
}
}

else

printf("multiplication not possible\n");
}

```

OUTPUT

1. Enter the no of rows and columns of matrix a :

3 3

Enter the no of rows and columns of matrix b :

3 3

Enter the elements of matrix a :

1 2 3 4 5 6 7 8 9

Enter the elements of matrix b :

1 2 3 4 5 6 7 8 9

First matrix a

1 2 3

4 5 6

7 8 9

Second matrix

1 2 3

4 5 6

7 8 9

Resultant matrix

14 32 50

32 77 122

50 122 194

2. Enter the no of rows and columns of matrix a :

2 2

Enter the no of rows and columns of matrix b :

3 3

Enter the elements of matrix a :

1 2 3 4

Enter the elements of matrix b :

1 2 3 4 5 6 7 8 9

First matrix a

1 2

4 5

Second matrix

1 2 3

4 5 6

7 8 9

Multiplication is not possible.

9. Develop a Program to compute Sin(x) using Taylor series approximation. Compare your result with the built- in Library function. Print both the results with appropriate messages.

```
#include<stdio.h>

#include<math.h>

void main()

{

float i=2,sum=0,term,denominator,numerator,x,degree;

printf("Enter the value of degree");

scanf("%f",&degree);

x=degree*(3.14/180);

numerator=x;

denominator=1;

do

{

term=numerator/denominator;

numerator=-numerator*x*x;

denominator=denominator*i*(i+1);

sum=sum+term;

i=i+2;

}

while(fabs(term)>=0.000001);

printf("\n Taylor series value of sin(%f) is %0.4f",degree,sum);

printf("\n According to library function sin(%f) is %0.4f\n",degree,sin(x));

}
```

OUTPUT:

1. Enter the value of degree : 30

Taylor series value of sin(30) is 0.4998

According to library function sin(30) is 0.4998

2. Enter the value of degree : 90

Taylor series value of sin(30) is 1.0000

According to library function sin(90) is 1.0000

10. Write functions to implement string operations such as compare, concatenate, string length. Convince the parameter passing techniques.

```
#include<stdio.h>

void strconcatinate(char str1[20],char str2[20]);

void strcmpare(char str1[20],char str2[20]);

void strlength(char str1[20]);

void main()

{

char str1[20],str2[20];

printf("Enter th first string\n");

scanf("%s",str1);

printf("Enter th second string\n");

scanf("%s",str2);

strconcatinate(str1,str2);

strcmpare(str1,str2);

strlength(str1);

strlength(str2);

}

void strconcatinate(char str1[20],char str2[20])

{

char finalstr[30];

int i,j;

for(i=0;str1[i]!='\0';i++)

finalstr[i]=str1[i];

finalstr[i]=' ';

for(j=0;str2[j]!='\0';j++)

finalstr[i+j+1]=str2[j];

finalstr[i+j+1]='\0';

printf("After concatination: %s\n",finalstr);
```

```

}

void strcmpare(char str1[20],char str2[20])
{
char finalstr[30];

int i=0;

while(str1[i]==str2[i]&&str1[i]!='\0'&&str2[i]!='\0')
i=i+1;

if(str1[i]!='\0'&&str2[i]!='\0')
printf("Strings are equal\n");

else
printf("Strings are not equal\n");
}

void strlenlength(char str1[20])
{
int i;

for(i=0;str1[i]!='\0';i++);

printf("Length of the string is %d\n",i);
}

```

OUTPUT:

Enter the first string : smvitm

Enter the second string: bantakal

After the concatinare : smvitm bantakal

Strings are not equal

Length of the string is 6

Length of the string is 8

11. Develop a program to sort the given set of N numbers using Bubble sort.

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

```
main()
```

```
{
```

```
int n,a[20],i,temp,j;
```

```
printf("Enter the number of elements:\n");
```

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

```
printf("Enter the elements:\n");
```

```
for(i=0;i<n;i++)
```

```
{
```

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

```
}
```

```
printf("Array before sorting:\n");
```

```
for(i=0;i<n;i++)
```

```
{
```

```
printf("%d\t",a[i]);
```

```
}
```

```
for(i=0;i<=n-1;i++)
```

```
{
```

```
for(j=0;j<=n-1-i;j++)
```

```
{
```

```
if(a[j]<a[j+1])
```

```
{
```

```
temp=a[j];
```

```
a[j]=a[j+1];
```

```
a[j+1]=temp;
```

```
}
```

```
}
```

```
}
```

```
printf("Elements after sorting:\n");  
for(i=0;i<n;i++)  
{  
printf("%d\t",a[i]);  
}  
printf("\n");  
}
```

OUTPUT:

Enter the number of elements: 5

Enter the elements: 7 2 8 3 4

Array before sorting: 7 2 8 3 4

Elements after sorting:8 7 4 3 2

12. Develop a program to find the square root of a given number N and execute for all possible inputs with appropriate messages. Note: Don't use library function sqrt(n).

```
#include<stdio.h>

void main()
{
    int i,n;
    float j;
    printf("Enter the number\n");
    scanf("%d",&n);
    j=n;
    for(i=1;i<=2*n;i++)
    {
        j=(j+(n/j))/2.0;
    }
    printf("square root of %d is%0.3f\n",n,j);
}
```

OUTPUT:

1. Enter the number : 4

Square root of 4 is 2.000

2. Enter the number : 6

Square root of 6 is 2.449

13. Implement structures to read, write and compute average- marks and the students scoring above and below the average marks for a class of N students.

```
#include<stdio_ext.h>

struct student
{
    int usn;
    char name[20];
    float marks;
}st[20];

void main()
{
    int n,i;
    float avgmarks,totmarks=0.0;
    printf("Enter the number of student\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nEnter the details of student:%d\n",i+1);
        printf("\nEnter USN:\n");
        scanf("%d",&st[i].usn);
        printf("\n Enter the name:\n");
        scanf("%s",st[i].name);
        printf("\nEnter the marks:\n");
        scanf("%f",&st[i].marks);
    }
    printf("Students details are:\n-----\n");
    printf("\n usn\t name\t marks\t\n");
    printf("-----");
    for(i=0;i<n;i++)
```



```

{
printf("\n\n%d\t%s\t%0.2f\n",st[i].usn,st[i].name,st[i].marks);
}
for(i=0;i<n;i++)
{
totmarks=totmarks+st[i].marks;
}
avgmarks=totmarks/n;
printf("\nAverage mark is %0.2f\n",avgmarks);
int m=0,l=0;
for(i=0;i<n;i++)
{
if(st[i].marks>=avgmarks)
{
m++;
}
else{
l++;
}
}

printf("\n %dStudents are having marks more than avgmarks\n%d Students are having marks less
than avgmarks\n",m,l);
}

```

OUTPUT:

Enter the number of student: 2

Enter the details of student: 1

Enter USN : 11

Enter the Name : ravi

Enter marks : 66

Enter USN : 12

Enter the Name : ram

Enter marks : 55

Student details are :

usn	name	marks

11	ravi	66
12	ram	55

Average mark is : 60.5

1 Students are having marks more than avgmarks

1 Students are having marks less than avgmarks

14. Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

```
#include<stdio.h>

#include<math.h>

void main()

{

float a[10],*ptr,mean,std,sum=0,sumstd=0;

int n,i;

printf("enter the number of elements:\n");

scanf("%d",&n);

printf("enter the array elements\n");

for(i=0;i<n;i++)

{

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

}

ptr=a;

for(i=0;i<n;i++)

{

sum=sum+*ptr;

ptr++;

}

mean=sum/n;

ptr=a;

for(i=0;i<n;i++)

{

sumstd=sumstd+pow((*ptr-mean),2);

ptr++;

}

std=sqrt(sumstd/n);
```

```
printf("sum=%f\n",sum);  
printf("mean=%f\n",mean);  
printf("standard deviation=%f\n",std);  
}
```

OUTPUT:

Enter the number of elements: 5

Enter the array elements : 1 5 6 7 9

Sum : 28.0000

Mean : 5.6000

Standard deviation : 2.65330

15. Implement Recursive functions for Binary to Decimal Conversion.

```
#include<stdio.h>

int bintodec(int n);

main()
{
    int decnum,n;
    printf("Enter any binary no.:\n");
    scanf("%d",&n);
    decnum=bintodec(n);
    printf("Decimal equivalent is %d\n",decnum);
}

int bintodec(int n)
{
    if (n==0)
        return 0;
    return (n%10+2*bintodec(n/10));
}
```

OUTPUT:

Enter any binary no. = 101

Decimal equivalent = 5