

# SVR ENGINEERING COLLEGE

AYYALURUMETTA (V), NANDYAL, KURNOOL DT.  
ANDHRA PRADESH – 518502



2019-2020

## LABORATORY MANUAL

OF

**Problem Solving & Programming Lab  
(19A05101P)**

**(R-19 REGULATION)**

Prepared by

**Mr. S.ASADULLA**

Asst. Professor

For

**B.Tech I YEAR – I SEM. (CSE)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SVR ENGINEERING COLLEGE**

(AFFILIATED TO JNTUA ANANTHAPURAM- AICITE-INDIA)  
AYYALURUMETTA (V), NANDYAL, KURNOOL DT.  
ANDHRA PRADESH – 518502

## **LAB MANUAL CONTENT**

### **Problem Solving & Programming Lab(19A05101P)**

Institute Vision & Mission, Department Vision & Mission

1. PO, PEO & PSO Statements.
2. List of Experiments
3. CO-PO Attainment
4. Experiment Code and Outputs

#### **1. Institute Vision & Mission, Department Vision & Mission**

##### **Institute Vision:**

To produce Competent Engineering Graduates & Managers with a strong base of Technical & Managerial Knowledge and the Complementary Skills needed to be Successful Professional Engineers & Managers.

##### **Institute Mission:**

To fulfill the vision by imparting Quality Technical & Management Education to the Aspiring Students, by creating Effective Teaching/Learning Environment and providing State – of the – Art Infrastructure and Resources.

##### **Department Vision:**

To produce Industry ready Software Engineers to meet the challenges of 21st Century.

##### **Department Mission:**

- Impart core knowledge and necessary skills in Computer Science and Engineering through innovative teaching and learning methodology.
- Inculcate critical thinking, ethics, lifelong learning and creativity needed for industry and society.
- Cultivate the students with all-round competencies, for career, higher education and self-employability.

## 2. PO, PEO& PSO Statements

### PROGRAMME OUTCOMES (POs)

**PO-1: Engineering knowledge** - Apply the knowledge of mathematics, science, engineering fundamentals of Computer Science& Engineering to solve complex real-life engineering problems related to CSE.

**PO-2: Problem analysis** - Identify, formulate, review research literature, and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO-3: Design/development of solutions** - Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, cultural, societal and environmental considerations.

**PO-4: Conduct investigations of complex problems** - Use research-based knowledge and research methods, including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

**PO-5: Modern tool usage** - Select/Create and apply appropriate techniques, resources and modern engineering and IT tools and technologies for rapidly changing computing needs, including prediction and modeling to complex engineering activities, with an understanding of the limitations.

**PO-6: The engineer and society** - Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CSE professional engineering practice.

**PO-7: Environment and Sustainability** - Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

**PO-8: Ethics** - Apply ethical principles and commit to professional ethics and responsibilities and norms of the relevant engineering practices.

**PO-9: Individual and team work** - Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO-10: Communication** - Communicate effectively on complex engineering activities with the engineering community and with the society-at-large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

**PO-11: Project management and finance** - Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO-12: Life-long learning** - Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological changes.

## Program Educational Objectives (PEOs):

**PEO 1:** Graduates will be prepared for analyzing, designing, developing and testing the software solutions and products with creativity and sustainability.

**PEO 2:** Graduates will be skilled in the use of modern tools for critical problem solving and analyzing industrial and societal requirements.

**PEO 3:** Graduates will be prepared with managerial and leadership skills for career and starting up own firms.

## Program Specific Outcomes (PSOs):

**PSO 1:** Develop creative solutions by adapting emerging technologies / tools for real time applications.

**PSO 2:** Apply the acquired knowledge to develop software solutions and innovative mobile apps for various automation applications

## Subject Time Table

SVR ENGINEERING COLLEGE::NANDYAL									
DEPARTMENT OF CSE									
S.ASADULLA					II-I				
Day/ Time	9:30 AM	10:20 AM	11:30 AM	12:20 PM-	LUNCH BREAK	02:00 PM	02:50 PM	03:40 PM	
	10:20 AM	11:10AM	12:20 PM	01:10 PM		02:50 PM	03:40 PM	04:30 PM	
MON									
TUE									
WED							PSP LAB		
THU									
FRI									
SAT									

SVR ENGINEERING COLLEGE				
Department:		COMPUTER SCIENCE & ENGINEERING		
Course Outcome Attainment - Internal Assessments				
Name of the faculty :		Mr.S.Asadulla	Academic Year:	2019-20
Branch & Section:		CSE	Exam:	EXTERNAL LAB
Course:		PROBLEM SOLVING & PROGRAMMING Lab	Semester:	I-I SEM
<b>Course Outcomes</b>	<b>Internal Lab</b>			<b>University Exam</b>
19A05101.1	3			3
19A05101.2	3			3
19A05101.3	3			3
19A05101.4	3			3
19A05101.5	3			3
Course Outcomes				Attainment Level
19A05101P.1	Construct his own computer using parts			<b>3.00</b>
19A05101P.2	Recognize the importance of programming language independent constructs			<b>3.00</b>
19A05101P.3	Solve computational problems			<b>3.00</b>
19A05101P.4	Select the features of C language appropriate for solving a problem			<b>3.00</b>
19A05101P.5	Design computer programs for real world problems			<b>3.00</b>
Average Attainment				<b>3.00</b>
<b>Overall Course Attainment</b>				<b>3.00</b>

<b>SVR ENGINEERING COLLEGE</b>														
<b>DEPARTMENT</b>		<b>COMPUTER SCIENCE &amp; ENGINEERING</b>												
<b>PROGRAM OUTCOME ATTAINMENT</b>														
Name of Faculty:		Mr.S.Asadulla					Academic Year			2019-20				
Branch & Section:		CSE					SUB CODE:			19A05101P				
Course:		PROBLEM SOLVING & PROGRAMMING LAB					Semester:			I-I				
<b>COURSE OUTCOME ATTAINMENT</b>														
Course outcome attainment			Internal lab	External lab										
19A05101 .1			3	3										
19A05101 .2			3	3										
19A05101 .3			3	3										
19A05101 .4			3	3										
19A05101 .5			3	3										
Average of CO Attainment			3	3										
<b>Attainment of the Course</b>			3											
<b>COURSE OUTCOMES AND PROGRAM OUTCOMES MAPPING</b>														
	PO 1	PO 2	PO3	PO4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2
19A05101 P.1	3	3	2	2	1	2			1		1	2	3	1
19A05101 P.2	3	2	1	1		1		1				1	2	2
19A05101 P.3	3	3	2	2	2	1	1			1			2	1
19A05101 P.4	3	3	1	2		2			2		2	2	2	2
19A05101 P.5	3	2	2	2		2		2		1			3	1
Average	3.0	2.6	1.6	1.8	1.5	1.6	1.0	1.5	1.5	1.0	1.5	1.7	2.4	1.4
<b>Overall Attainment of the Course</b>	<b>3</b>	<b>2.6</b>	<b>1.6</b>	<b>1.8</b>	<b>1.5</b>	<b>1.6</b>	<b>1</b>	<b>1.5</b>	<b>1.5</b>	<b>1</b>	<b>1.5</b>	<b>1.67</b>	<b>2.4</b>	<b>1.4</b>
Faculty Mr.S.Asadulla					Head of the Department									

## LIST OF EXPERIMENTS (SYLLABUS)

### JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

#### B.Tech – I-I Sem

#### (19A05101P) Problem Solving & Programming Lab

#### (Common to CSE & IT)

#### Objectives:

- □ Learn C Programming language
- □ To make the student solve problems, implement algorithms using C language

#### Laboratory Experiments #

1. Assemble and disassemble parts of a Computer
2. Design a C program which reverses the number
3. Design a C program which finds the second maximum number among the given list of numbers.
4. Construct a program which finds the kth smallest number among the given list of numbers.
5. Design an algorithm and implement using C language the following exchanges a □ b □ c □ d
6. Develop a C Program which counts the number of positive and negative numbers separately and also compute the sum of them.
7. Implement the C program which computes the sum of the first n terms of the series  $Sum = 1 - 3 + 5 - 7 + \dots$
8. Design a C program which determines the numbers whose factorial values are between 5000 and 32565.
9. Design an algorithm and implement using a C program which finds the sum of the infinite series  
 $1 - x^2/2! + x^4/4! - x^6/6! + \dots$
10. Design a C program to print the sequence of numbers in which each number is the sum of the three most recent predecessors. Assume first three numbers as 0, 1, and 1.
11. Implement a C program which converts a hexadecimal, octal and binary number to decimal number and vice versa.
12. Develop an algorithm which computes the all the factors between 1 to 100 for a given number and implement it using C.
13. Construct an algorithm which computes the sum of the factorials of numbers between m and n.
14. Design a C program which reverses the elements of the array.
15. Given a list of n numbers, Design an algorithm which prints the number of stars equivalent to the value of the number. The stars for each number should be printed horizontally.
16. Implement the sorting algorithms a. Insertion sort b. Exchange sort c. Selection sort d. Partitioning sort.
17. Illustrate the use of auto, static, register and external variables.
18. Design algorithm and implement the operations creation, insertion, deletion, traversing on a singly linked list.
19. Develop a C program which takes two numbers as command line arguments and finds all the

common factors of those two numbers.

20. Design a C program which sorts the strings using array of pointers.

# The above list is not exhaustive. Instructors may add some experiments to the above list. Moreover, 50% of the experiments are to be changed every academic year. Instructors can choose the experiments, provided those experiments are not repetitions.

**Course outcomes:** Student should be able to

1. Construct a Computer given its parts (L6)
2. Select the right control structure for solving the problem (L6)
3. Analyze different sorting algorithms (L4)
4. Design solutions for computational problems (L6)
5. Develop C programs which utilize the memory efficiently using programming constructs like pointers.

**References:**

1. B. Govindarajulu, "IBM PC and Clones Hardware Trouble shooting and Maintenance", Tata McGraw-Hill, 2<sup>nd</sup> edition, 2002.
2. R.G. Dromey, "How to Solve it by Computer". 2014, Pearson.

1. Assemble and disassemble parts of a Computer

**AIM:** Disassemble and assemble the PC back to working condition. Students should be able to trouble shoot the computer and identify working and non-working parts. Student should identify the problem correctly by various methods available (eg: beeps). Students should record the process of assembling and trouble shooting a computer.

**PROCEDURE:** List of Computer Parts are:

1. Cabinet,
2. Mother Board,
3. Processor,
4. Hard Disk,
5. RAM,
6. SMPS,
7. Floppy Drive,
8. DVD R/W Drive,
9. AGP Card, and
10. VGA Card etc...

**MATERIALS REQUIRED:**

1. Philips head Screwdriver (also known as Star Screwdriver)
2. Flat head Screwdriver
3. Forceps (for pulling out jumpers and screws)
4. Magnetized Screwdriver
5. Multi meter (Testing)

**DISASSEMBLE:** Disassembling means removing part by part in a system or removing the parts

separately.

#### **STEPS TO DISASSEMBLE A PC:-**

- 1) Remove the Power Supply from wall socket and remove all the connections.
- 2) Remove all the screws of the cabinet with a star headed screw driver and open the cabinet doors.
- 3) Remove the connections to the processor from SMPS.
- 4) Remove the SMPS
- 5) Remove the Heat Sink fan which is placed on the top of the processor.
- 6) Remove the Processor by raising the lift lever.
- 7) Remove the RAM from the RAM Slots.
- 8) Remove the AGP Cards if any.
- 9) Also Remove VGA Cards if any.
- 10) Remove the Network Cards if any.
- 11) Remove the Connections to the Hard Disk from motherboard.
- 12) Remove the Connections to the CD/DVD/Floppy Disk from motherboard.
- 13) Remove the Motherboard from the cabinet.

**ASSEMBLE:** After buying all the computer components they must be put together. This process is called assembling.

#### **STEPS TO ASSEMBLE A PC:-**

- 1) Install the Motherboard to the cabinet using the star headed screw driver.
- 2) Install the Processor by raising the lift lever on the CPU Socket.
- 3) Install the Heat Sink fan.
- 4) Install the SMPS (Switch Mode Power Supply)
- 5) Give the connections to the processor from the SMPS
- 6) Network Cards.
- 7) Install the Hard Disk.
- 8) Install the CD/DVD/Floppy Disk to the motherboard.
- 9) Screw the doors of the Install the RAM in to the RAM Slots
- 10) Install the VGA Cards. (VGA à Video Graphic Adapter)
- 11) Install the AGP Cards. (AGP à Accelerated Graphics Port)
- 12) Install the cabinet using the star headed screw driver.
- 13) Finally give all the connections including the power supply from the wall socket.

NOTE: Be sure to handle all the components with great care. If a small thing like a screw is dropped on the Motherboard; it can damage the delicate circuitry.

## **2.. Write a program to reverse the digits of a number.**

**Aim :-** to write a program to reverse the digits of a number

#### **Program Description:-**

In this program , for each iteration we divide the number by 10 and the remainder is added to a variable sum which is multiplied by 10 for every iteration and we reduce the number to quotient. The process repeats until  $n > 0$

**Source Code:**

```
#include<stdio.h>
#include<conio.h>
void main(
{
    int n,r,s=0;
    clrscr();
    printf("\n Enter a number :");
    scanf("%d",&n);
    while(n>0)
    {
        r=n%10;
        s=(s*10)+r;
        n=n/10;
    }
    printf("\n The reverse of given number is:%d",s);
}
```

**Input:**

Enter a number :325

**Output:**

The reverse of given number is:523

**Viva Questions**

1. When we storing sum in a variable ,that variable is assigned with the value zero .why
2. What is a remainder operator.
3. Can we apply % operator for float and double values.
4. When we storing product in a variable , that variable is assigned with the value one. Why
5. What is the use of void.

**3. Design a C program which finds the second maximum number among the given list of numbers.**

**Program Description:-** In this program first we are reading size of list and entering elements into the list. For finding second maximum element list is sorted in descending order and from the list second element is printed.

```
#include<stdio.h>
Void main()
{
int a[100],n,i,t,j;
printf("Enter size of the list\n");
scanf("%d",&n);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
```

```

for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if(a[i]<a[j])
{
t=a[i];
a[i]=a[j];
a[j]=t;
}
for(i=0;i<n;i++)
printf("%d\t",a[i]);
Printf("Second maximum number among the given list of numbers is %d",a[1]);
}

```

**Input:** Enter size of the list:

5  
7  
10  
13  
8  
4

**Output:**

13 10 8 7 5

Second maximum number among the given list of numbers is 10

4. Construct a program which finds the kth smallest number among the given list of numbers.

**Program Description:-** In this program first we are reading size of list and entering elements into the list. For finding k<sup>th</sup> smallest element, list is sorted in ascending order and from the list k<sup>th</sup> smallest element is printed.

```

#include<stdio.h>
Void main()
{
int a[100],n,i,t,j,k;
printf("Enter size of the list\n");
scanf("%d",&n);
printf("enter the value of k");
scanf("%d",&k);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if(a[i]>a[j])
{
t=a[i];

```

```

a[i]=a[j];
a[j]=t;
}
for(i=0;i<n;i++)
printf("%d\t",a[i]);
Printf(" %d smallest number among the given list of numbers is %d",k,a[k-1]);
}

```

**Input:** Enter size of the list:

```

5
enter the value of k 3
7
10
13
8
4

```

**Output:**

```

5 7 8 10 13
3 smallest number among the given list of numbers is 8

```

### Viva Questions

1. In a two dimensional array , what the first and second subscripts represents.
2. What is the difference between structure and an array
3. Can we produce the array size at run time. Justify
4. What are different dynamic memory allocation functions.

What do you mean by dynamic initialization of a variable

### **5.Design an algorithm and implement using C language the following exchanges**

$a \leftarrow b \leftarrow c \leftarrow d$

**Program Description:** we are taking four values and swapping them with the help of a temporary variable

```

#include<stdio.h>
void main()
{
int a,b,c,d,t;
printf("enter four values");
scanf("%d%d%d%d",&a,&b,&c,&d);
t=a;
a=b;
b=c;
c=d;
d=t;
}

```

```
printf("a=%d,b=%d,c=%d, d=%d",a,b,c,d);  
}
```

**Output:**

Input:

enter four values 1 2 3 4

Output:

a=2 b=3 c=4 d=1

**Viva Questions**

1. What is initialization?
2. What are input and output statements?
3. What is datatype?
4. How many main() functions does a program have?
5. What is declaration?

6. Develop a C Program which counts the number of positive and negative numbers separately and also compute the sum of them.

Program Description:

In this program we are reading size of list and reading both positive and negative numbers. we are printing number of positive and negative numbers and their sum.

```
#include<stdio.h>  
  
void main()  
{  
int a[100],i,j,pc=0,nc=0,ps=0,ns=0,n;  
printf("enter size of list");  
scanf("%d",&n);  
printf("enter elements into list");  
for(i=0;i<n;i++)  
scanf("%d",&a[i]);  
for(i=0;i<n;i++)
```

```

{
if(a[i]>0)
{
pc++;
ps=ps+a[i];
}
else{
nc++;
ns=ns+a[i];
}
}

printf("number of positive elements=%d sum of positive elements is %d",pc,ps);
printf("number of negative elements=%d sum of negative elements is %d",nc,ns);
}

```

Output:

enter size of list:9

enter elements into list:

8 -5 4 2 -7 -23 65 -32 24

Output:-

number of positive elements=5 sum of positive elements is: 103

number of negative elements=4 sum of negative elements is:-67

Viva Questions

1. int a[20];

After declaring the above array , if we use only 'a'(without subscript) , what it refers to.

2. What do you mean by static memory allocation.

3. What is the difference between prefix and postfix increment.

4. What is the difference between unary and binary operators.

5. What are membership operators

7. Implement the C program which computes the sum of the first n terms of the series

Sum = 1 - 3 + 5 - 7 + 9

**Program Description:** In this program we are reading number of terms and we are printing the sum .

```
#include<stdio.h>

void main()
{
int i,n,sum=0,j;
printf("enter value of n");
scanf("%d",&n);
j=1;
for(i=1;i<n;i++)
{
if(i%2!=0)
sum=sum+j;
else
sum=sum-j;
j=j+2;
}
printf("sum of first n numbers=%d",sum);
}
```

Output:

Input:

enter value of n 5

Output:

sum of first n numbers=5

### Viva Questions

1. What is the lifetime of a variable.
2. What are storage classes.
3. Which header file to include in order to use mathematical functions.
4. What is the difference between getc(), getch() and getche()
5. What is difference between keyword and reserved word.

### **8.Design a C program which determines the numbers whose factorial values are between 5000 and 32565.**

#### **Program Description:-**

In this program , we multiply a variable i whose value is 1 with another variable f which is initialized with 1. For each iteration we increment the value of a variable i by 1 and then multiply with same variable f. The process repeats until the  $i \leq n$ .

Source Code :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,f=1,i=1;
    clrscr();
    printf("\n Enter a number:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        j=i;
        k=1;
        f=1;
        while(k<=j)
        {
            f*=k;
            k++;
        }
        if(k>=5000 and k<=32565)
            printf(" %3d = ",j);
    }
}
```

```
        getch();
    }
}
```

Input

Enter a number:10

Output

9 10

**Viva Questions:**

1. What are looping statements
2. What is a definite loop. Specify an example
3. What is an indefinite loop. Specify with an example
4. What is the difference between while and do-while
5. What are minimum number of iterations in while and do-while

**9.Design an algorithm and implement using a C program which finds the sum of the infinite series  $1 - x^2/2! + x^4/4! - x^6/6! + \dots$**

**Program Description:** In this program we are reading number of terms, value of x and we are printing the sum of the series  $1 - x^2/2! + x^4/4! - x^6/6! + \dots$

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

int main()
{
    double x, sum = 1, term = 1, fact, j, y = 2, m;

    int n ,i;
    printf("enter the value of x and n");
    scanf("%lf%d",&x,&n);
```

```

for (i = 1; i < n; i++) {
    fact = 1;
    for (j = 1; j <= y; j++) {
        fact = fact * j;
    }
    term = term * (-1);
    m = term * pow(x, y) / fact;
    sum = sum + m;
    y += 2;
}
printf("Sum of series=%.2lf",sum);
}

```

Input : x = 9, n = 10

Output : -5.1463

Input : x = 5, n = 15

Output : 0.2837

Viva Questions:

1. What is the difference between entry-controlled loop and exit-controlled loop
2. What is the default scope of if – statement
3. What is the difference between initialization and assigning
4. What is the difference between = and ==
5. If(k) . Is this a valid condition. If yes .Specify the reason

**10 Design a C program to print the sequence of numbers in which each number is the sum of the three most recent predecessors. Assume first three numbers as 0, 1, and 1**

**Program Description:-**

In this program , we initialize variable f1 to zero, f2 to 1 and f3 to 1. We print the variable f which is the sum of f1, f2,f3. Then we interchange the values such that f3 is stored in f2 and f2 is stored in f1 and f value is stored in f3 . The process repeats until n elements are generated.

Source Code :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,f1=0,f2=1,f3=1,f;
    clrscr();
    printf(" \n Enter how many numbers you want to print : ");
    scanf("%d",&n);
    printf("\n series is :\n");
    scanf("%d\t%d",f1,f2,f3);
    for(i=4;i<=n;i++)
    {
        f=f1+f2+f3;
        printf("\t%d",f);
        f1=f2;
        f2=f3;
        f3=f;
    }
    getch();
}
```

Input:

Enter how many numbers you want to print : 8

Output:-

series is :

0 1 1 2 4 7 13 24

### Viva Questions

1. What are backslash Characters
2. List some backslash Characters
3. When to use for and when to use while
4. Can we write a for statement without initialization , condition and incrementation / decrementation.
5. What are infinite loops

**11. Implement a C program which converts a hexadecimal, octal and binary number to decimal number and vice versa.**

Aim:- To perform Base (decimal, octal, hexadecimal, etc) conversion

Program description:

In this program we input a decimal value and we convert that decimal value into binary, octal and hexa decimal value.

Source Code:

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

```
void dec_bin(int decimal)
{
    int r,q,bin[20],i=1,j;
    q=decimal;
    while(q!=0)
    {
        bin[i++]=q%2;
        q=q/2;
    }
    for(j=i-1;j>0;j--)
        printf("%d",bin[j]);
}
```

```
void dec_oct(int decimal)
{
    int r,q,oct[10],i=1,j;
    q=decimal;
    while(q!=0)
    {
        oct[i++] = q% 8;
        q=q/8;
    }
    for(j=i-1;j>0;j--)
        printf("%d",oct[j]);
}
```

```
void dec_hex(int decimal)
{
    int r,q,i=1,j,temp;
    char hexa[10];
    q=decimal;
    while(q!=0)
    {
        temp = q % 16;
```

```

        //To convert integer into character
        if( temp < 10)
            temp =temp + 48;
        else
            temp = temp + 55;

        hexa[i++]= temp;
        q=q/16;
    }
    for(j=i-1;j>0;j--)
        printf("%c",hexa[j]);
}

void main()
{
    int decimal,octal;
    char hexa[20];
    long int bin;
    clrscr();
    printf("\n Enter a decimal value:");
    scanf("%d",&decimal);
    printf("\n Given decimal value is=%d",decimal);
    printf("\n Equivalent binary value:");
    dec_bin(decimal);
    printf("\n Equivalent Octal value:");
    dec_oct(decimal);
    printf("\n Equivalent Hexa decimal value:");
    dec_hex(decimal);
}

```

### Output

```

Enter a decimal value:188
Given decimal value is=188
Equivalent binary value:10111100
Equivalent Octal value:274
Equivalent Hexa decimal value:BC

```

### Viva Questions

1. If a value is preceded with '0X' , it can be treated as which type of value.
2. If a value is preceded with '0' , it can be treated as which type of values.
3. If a value is preceded with U,F .what do you mean by that
4. When to use %u format specifier

5. What is the difference between signed and unsigned values.

**12.** Develop an algorithm which computes the all the factors between 1 to 100 for a given number and implement it using C.

**Program description:** In this program we are displaying all the factors of a number if it lies between 1 and 100.

```
#include<stdio.h>
void main()
{
int n,i;
printf("enter the value of n\n");
scanf("%d",&n);
printf("Factors\n");
for(i=1;i<=n,i++)
if((n%i==0) &&(i>=1 && i<=100))
printf("%d",i);
}
```

Input:

Enter a number :16

Output:

Factors

1 2 4 8 16

### Viva Questions

1. What are tokens.
2. What is the range of integer data type.
3. What are different data types available in c
4. What are preprocessor directives
5. What is the use of "include" preprocessor directive

---

**13.. Construct an algorithm which computes the sum of the factorials of numbers between**

## **m and n.**

### **Program description:**

In this program we are reading two num numbers and displaying sum of all factorials of numbers between m and n

```
#include<stdio.h>
Void main()
{
int m,n,i,j,fact,sum=0;
printf("Enter the values of m and n");
scanf("%d %d",&m,&n);
for(i=m;i<=n;i++)
{
fact=1;
for(j=1;j<=m;j++)
fact=fact*j;
sum=sum+fact;
}
printf("sum of factorials between m=%d and n=%d is %d",m,n,sum);
}
```

Input :

Enter the values of m and n

2 4

Output:

sum of factorials between m=2 and n=4 is 32

### **Viva Questions**

1. What do you mean by nesting of structures.
2. Why we should use & in case of scanf() and why we are not using & in case of printf()
3. Which header file to be included to implement clrscr().
4. Why we have to include an header file.
5. In c which is the terminating(or)delimiter operator.

14. Design a C program which reverses the elements of the array.

Program Description:-

In this we are writing a C program which reverses the elements of the array.

Source Code :-

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[50],i,n;
    clrscr();
    printf("\n Enter no. of elements:");
    scanf("%d",&n);
    printf("\n Enter the elements :");
    for(i=1;i<=n;i++)
        scanf("%d",&a[i]);

    printf("\n Given elements are:\n");
    for(i=1;i<=n;i++)
        printf("%5d",a[i]);
    printf("\n elements in reverse are:\n");
    for(i=n;i>=1;i--)
        printf("%5d",a[i]);

    getch();
}

```

#### Input

Enter no. of elements:6

Enter the elements : 12 10 8 2 24 15

#### Output

Given elements are:

12 10 8 2 24 15

elements in reverse are:

15 24 2 8 10 12

#### Viva Questions

1. What is an array
2. Array is static. Explain why
3. What are the advantages and disadvantages of an array.
4. What is a subscript.
5. What is the default range of a subscript.

**15..Given a list of n numbers, Design an algorithm which prints the number of stars equivalent to the value of the number. The starts for each number should be printed horizontally.**

**Program Description:-**

In this we are writing a C program which reads size of list and reads elements of list we are using nested loops to display number of stars depending on the value in the list.

Source Code :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[50],i,n,j;
    clrscr();
    printf("\n Enter no. of elements:");
    scanf("%d",&n);
    printf("\n Enter the elements :");
    for(i=1;i<=n;i++)
        scanf("%d",&a[i]);

    printf("\n Given elements are:\n");
    for(i=1;i<=n;i++)
        printf("%5d",a[i]);

    for(i=1;i<=n;i++)
        for(j=1;j<=a[i];j++)
            printf("\t *",j);

    getch();
}
```

**Input:**

Enter no. of elements: 4  
Enter the elements : 2 3 4 5

**Output:**

Given elements are:  
2 3 4 5

\*\*  
\*\*\*

\*\*\*\*  
\*\*\*\*\*

### Viva Questions

1. What is nesting of loops?
2. What is the use of break statement?
3. What is the use of continue statement?
4. Can we use conditional statements inside loops?
5. What is the use of getch()?

### 16. Implement the sorting algorithms

a. Insertion sort b. Exchange sort c. Selection sort d. Partitioning sort.

(a) Insertion sort

```
#include<stdio.h>
#include<conio.h>
void inst_sort(int []);
void main()
{
    int num[5],count;
    clrscr();
    printf("\nEnter the Five Elements to sort:\n");

    for (count=0;count<5;count++)
        scanf("%d",&num[count]);
    inst_sort(num);

    printf("\n\nElements after sorting: \n");
    for(count=0;count<5;count++)
        printf("%d\n",num[count]);
    getch();
}

// Function for Insertion Sorting
void inst_sort(int num[])
{
    int i,j,k;
    for(j=1;j<5;j++)
    {
        k=num[j];
        for(i=j-1;i>=0 && k<num[i];i--)
```

```
        num[i+1]=num[i];
        num[i+1]=k;
    }
}
```

INPUT & OUTPUT :

Enter the Five Elements to sort:

56  
43  
32  
23  
90

Elements after sorting:

23  
32  
43  
56  
90

b. Exchange sort

```
#include<stdio.h>
#define MAXSIZE 100
int a[MAXSIZE],n;
main()
{
    int i;
    clrscr();
    printf("How many elements you want to sort : ");
    scanf("%d",&n);
```

```

printf("Enter %d Elements\n",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]);
printf("\nArray After Bubble Sorting\n");
bubblesort(a,n);
for(i=0;i<n;i++)
    printf("%d\t",a[i]);
getch();
}

```

```

bubblesort(int a[],int n)
{
    int i,j,t;
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)

            if(a[i]>a[j])
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
}

```

INPUT & OUTPUT :

```

How many elements you want to sort : 5
Enter 5 Elements
10 4 5 67 90
Array before Sorting
10 4 5 67 90
Array After Bubble Sorting
4 5 10 67 90

```

c. Selection sort

```

#include<stdio.h>
int main()
{
    int array[100], n, c, d, p, swap;
    clrscr();
    printf("Enter number of elements\n");

```

```

scanf("%d", &n);

printf("Enter %d integers\n", n);

for ( c = 0 ; c < n ; c++ )
    scanf("%d", &array[c]);

for ( c = 0 ; c < ( n - 1 ) ; c++ )
{
    p = c;

    for ( d = c + 1 ; d < n ; d++ )
    {
        if ( array[p] > array[d] )
            p = d;
    }
    if ( p != c )
    {
        swap = array[c];
        array[c] = array[p];
        array[p] = swap;
    }
}

printf("Sorted list in ascending order:\n")
for ( c = 0 ; c < n ; c++ )
    printf("%d\t", array[c]);

return 0;
}

```

```

Enter number of elements
5
Enter 5 integers
1 40 3 50 6
Sorted list in ascending order:
1 3 6 40 50

```

d. Partitioning sort.

```

#include <stdio.h>
#define MAX 10

void swap(int *m,int *n)

```

```

{
    int temp;
    temp = *m;
    *m = *n;
    *n = temp;
}
int get_key_position(int x,int y )
{
    return((x+y) /2);
}

// Function for Quick Sort
void quicksort(int list[],int m,int n)
{
    int key,i,j,k;
    if( m < n)
    {
        k = get_key_position(m,n);
        swap(&list[m],&list[k]);
        key = list[m];
        i = m+1;
        j = n;
        while(i <= j)
        {
            while((i <= n) && (list[i] <= key))
                i++;
            while((j >= m) && (list[j] > key))
                j--;
            if( i < j)
                swap(&list[i],&list[j]);
        }
        swap(&list[m],&list[j]);
        quicksort(list,m,j-1);
        quicksort(list,j+1,n);
    }
}

// Function to read the data
void read_data(int list[],int n)
{
    int j;
    printf("\n\nEnter the elements:\n");
    for(j=0;j<n;j++)
        scanf("%d",&list[j]);
}

```

```

}

// Function to print the data
void print_data(int list[],int n)
{
    int j;
    for(j=0;j<n;j++)
        printf("%d\t",list[j]);
}

void main()
{
    int list[MAX], num;
    clrscr();
    printf("\n***** Enter the number of elements Maximum [10] *****\n");
    scanf("%d",&num);
    read_data(list,num);
    printf("\n\nElements in the list before sorting are:\n");
    print_data(list,num);
    quicksort(list,0,num-1);
    printf("\n\nElements in the list after sorting are:\n");
    print_data(list,num);
    getch();
}

```

## 11.6 INPUT & OUTPUT :

\*\*\*\*\* Enter the number of elements Maximum [10] \*\*\*\*\*

6

Enter the elements:

34

78

65

12

32

11

Elements in the list before sorting are:

34 78 65 12 32 11

Elements in the list after sorting are:

11 12 32 34 65 78

### Viva Questions

1. What is an in-place sort ?
2. Which sorting techniques comes under internal sorting ?
3. What is sorting?
4. Difference between internal and external sorting?
5. What is insertion sort?

### 17. Illustrate the use of auto, static, register and external variables.

Program Description :

In this program we will come to know about scope and lifetime of a variable

#### Auto variable:

```
int add()  
{  
    int a=13;  
    auto int b=48;  
    return a+b;  
}  
  
#include <stdio.h>  
int main( )  
{  
    auto int j = 1;  
    {  
        auto int j= 2;  
        {  
            auto int j = 3;  
            printf ( " %d ", j);  
        }  
        printf ( "\t %d ",j);  
    }  
    printf( "%d\n", j);  
}
```

Output:

3 2 1

**Static Variable:**

```
#include<stdio.h>
void main()
{
incr();
incr();
incr();
}
incr()
{
static int a;
a++;
printf(“%d”,a);
}
```

Output:

1 2 3

**Register Variable:**

```
main()
{
register int j;
for(j=0;j<5;j++)
printf(“%4d”,j);
}
```

Output:

0 1 2 3 4

**Extern Variables:**

```
#include<stdio.h>
void main()
{
extern int j;
incr();
incr();
```

```
incr();  
}  
incr()  
{  
j++;  
printf("%d",j);  
}
```

Output:  
1 2 3

### **Viva Questions**

1. What does a storage class specify?
2. What are automatic variables?
3. What are static variables?
4. What are register variables?
5. What are extern variables?

**18.Design algorithm and implement the operations creation, insertion, deletion,**

### traversing on a singly linked list.

#### Program Description:

In this program we are creating a self referential structure and initialize its members. we are implementing all functions to perform Operations on Singly linked list .

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
struct node
{
    int data;
    struct node *next;
};
struct node *start;
void main()
{
    int choice,n,e,position,i;
    clrscr();
    start=NULL;
    while(1)
    {
        printf("1.Create a linked List\n");
        printf("2.Add at begining\n");
        printf("3.Add after \n");
        printf("4.Delete\n");
        printf("5.Display\n");
        printf("6.Count\n");
        printf("7.Search\n");
        printf("8.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("How many nodes you want in the list: ");
                scanf("%d",&n);
                for(i=0;i<n;i++)
                {
                    printf("Enter the element : ");
                    scanf("%d",&e);
                    create(e);
                }
                break;
            case 2:
```

```

        printf("Enter the element : ");
        scanf("%d",&e);
        add_beg(e);
        break;
    case 3:
        printf("Enter the element : ");
        scanf("%d",&e);
        printf("Enter the position after which this element is inserted : ");
        scanf("%d",&position);
        add_after(e,position);
        break;
    case 4: if(start==NULL)
        {
            printf("List is empty\n");
            continue;
        }
        printf("Enter the element for deletion : ");
        scanf("%d",&e);
        del(e);
        break;
    case 5: display();
        break;
    case 6: count();
        break;
    case 7: printf("Enter the element to be searched : ");
        scanf("%d",&e);
        search(e);
        break;
    case 8: exit();
    default:
        printf("Wrong choice\n");
    } //End of switch
} //End of while
} //End of main()

create(int d)
{
    struct node *q,*temp;
    temp= malloc(sizeof(struct node));
    temp->data=d;
    temp->next=NULL;

    if(start==NULL) /*If list is empty */
        start=temp;
}

```

```

else
{ /*Element inserted at the end */
    q=start;
    while(q->next!=NULL)
    q=q->next;
    q->next=temp;
}
}
//End of create()
add_beg(int d)
{
    struct node *temp;
    temp=malloc(sizeof(struct node));
    temp->data=d;
    temp->next=start;
    start=temp;
} /*End of add_beg()*/
add_after(int d,int pos)
{
    struct node *temp,*q;
    int i;
    q=start;
    for(i=0;i<pos-1;i++)
    {
        q=q->next;
        if(q==NULL)
        {
            printf("There are less than %d elements",pos);
            return;
        }
    }
    temp=malloc(sizeof(struct node) );
    temp->next=q->next;
    temp->data=d;
    q->next=temp;
} //End of add_after()
del(int d)
{
    struct node *temp,*q;
    if(start->data == d)
    {
        temp=start;
        start=start->next; /*First element deleted*/
        free(temp);
    }
}

```

```

        return;
    }
    q=start;
    while(q->next->next != NULL)
    {
        if(q->next->data==d) /*Element deleted in between*/
        {
            temp=q->next;
            q->next=temp->next;
            free(temp);
            return;
        }
        q=q->next;
    } /*End of while */
    if(q->next->data==d) /*Last element deleted*/
    {
        temp=q->next;
        free(temp);
        q->next=NULL;
        return;
    }
    printf("Element %d not found\n",d);
} //End of del()

```

```

display()
{
    struct node *q;
    if(start == NULL)
    {
        printf("List is empty\n");
        return;
    }
    q=start;
    printf("List is :\n");
    while(q!=NULL)
    {
        printf("%d ", q->data);
        q=q->next;
    }
    printf("\n");
} //End of display()
count()
{
    struct node *q=start;

```

```

    int cnt=0;
    while(q!=NULL)
    {
        q=q->next;
        cnt++;
    }
    printf("Number of elements are %d\n",cnt);
} //End of count()
search(int d)
{
    struct node *ptr = start;
    int pos = 1;
    while(ptr!=NULL)
    {
        if(ptr->data==d)
        {
            printf("Item %d found at position %d\n",d,pos);
            return;
        }
        ptr = ptr->next;
        pos++;
    }
    if(ptr == NULL)
        printf("Item %d not found in list\n",d);
} //End of search()

```

INPUT & OUTPUT:

1. Create a linked List
2. Add at beginning
3. Add after
4. Delete
5. Display
6. Count
7. Search
8. Quit

Enter your choice : 1  
How many nodes you want in the list: 5  
Enter the element : 45  
Enter the element : 78  
Enter the element : 55  
Enter the element : 98

Enter the element : 12

- 1.Create a linked List
- 2.Add at begining
- 3.Add after
- 4.Delete
- 5.Display
- 6.Count
- 7.Search
- 8.Quit

Enter your choice : 2

Enter the element : 12

- 1.Create a linked List
- 2.Add at begining
- 3.Add after
- 4.Delete
- 5.Display
- 6.Count
- 7.Search
- 8.Quit

Enter your choice : 3

Enter the element : 7

Enter the position after which this element is inserted : 78

There are less than 78 elements1.Create a linked List

- 2.Add at begining
- 3.Add after
- 4.Delete
- 5.Display
- 6.Count
- 7.Search
- 8.Quit

Enter your choice : 5

List is :

12 45 78 55 98 12

- 1.Create a linked List
- 2.Add at begining
- 3.Add after
- 4.Delete
- 5.Display
- 6.Count
- 7.Search
- 8.Quit

Enter your choice :8

### **Viva Questions:**

1. What is data structure ?
2. What are the types of data structures ?
3. What is an array?
4. What are the disadvantages of array?
5. What is linked list?
6. What is the draw back with static memory allocation?
7. What are the advantages of linked list?
8. How to add an element in a linked list at first?
9. What is the logic for insertion of an element at middle ?
10. How to delete an element ?
11. What are head and null nodes ?
12. What are the drawbacks with singly linked list ?

**19.19. Develop a C program which takes two numbers as command line arguments and finds all the common factors of those two numbers.**

**Program Description:**

Main function takes two parameters one is argument count and the other is argument vector. We are passing two values to main function and displaying all common factors of those two numbers.

```
#include<stdio.h>
int main(int argc, char *argv[])
{
    int a, b, ;
    a = atoi(argv[1]);
    b = atoi(argv[2]);
    if(a<b)
    {
        for(j=1;j<=a;j++)
        if(a%j==0 && b%j==0)
        printf("%d\t", j);
    }
    else
    {
        for(j=1;j<=b;j++)
        if(a%j==0 && b%j==0)
        printf("%d\t", j);
    }
}
```

Input:  
12 24  
Output:

1 2 3 4 6 12

### Viva Questions:

1. What are command line arguments?
2. What is argc?
3. What is argv[]?
4. What are the properties of command line arguments?

### 20.Design a C program which sorts the strings using array of pointers

Aim:- To write a program to read a set of strings and sort them in alphabetical order

#### Program Description:-

The sorting of strings is performed in similar to number sorting where we compare the first number with all the remaining elements to identify the minimum element and placed it in first position. Similarly we identify all the elements and place them in the corresponding positions.

#### Source Code:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[15][20],temp[20];
    int i,j,n;
    clrscr();
    printf("\n Enter number of strings :");
    scanf("%d",&n);
    printf("\n Enter the strings:\n");
    for(i=1;i<=n;i++)
        scanf("%s",str[i]);
    for(i=1;i<n;i++)
        for(j=i+1;j<=n;j++)
            if(strcmp(str[i],str[j])>0)
            {
                strcpy(temp,str[i]);
                strcpy(str[i],str[j]);
                strcpy(str[j],temp);
            }
    printf("\n After sorting strings are :\n");
    for(i=1;i<=n;i++)
        printf("%s\n",str[i]);
    getch();
}
```

Input:-

Enter number of strings :6

Enter the strings

ramu

krishna

lakshman

balaji

deepak

abhiram

Output:-

After sorting strings are :

abhiram

balaji

deepak

krishna

lakshman

ramu

Viva Questions

1. How to store multiple strings
2. Can we copy one string to another without using strcpy()
3. What is the difference between strdup() and strcpy()
4. Write the structure of a string
5. What is the difference between 'a' and "a"
6. How to access string by using pointers.
7. What is the use of strrev()
8. What is the difference between upper case and lower case characters
9. C is a case sensitive language. Justify
10. What is the use of strlen()















