**ROOTS OF A QUADRATIC EQUATION**

**AIM** : Program to find the roots of a quadratic equation.

**INPUT SPECIFICATION :** a , b , c - coefficients of the quadratic equation

**OUTPUT SPECIFICATION :** Root1 , Root2 - Roots of quadratic equation

Printswhether the roos are equal , distict or complex

**ALGORITHM**

1. Start
2. Get the coefficients of the quadratic equation A, B and C
3. disc ← (B \* B) – (4 \* A \* C)
4. IF D = 0 THEN

Begin

Print “Roots are real and equal”

Root1 = -B / (2 \* A)

Root2 = Root1

Print “First root = ” , Root1

Print “Second root = ” , Root2

End

ELSE

Begin

IF D > 0 THEN

Begin

Print “Roots are real and distinct”

Root1 = ( -B + √disc ) /( 2 \* A)

Root2 = ( -B - √disc ) /( 2 \* A)

Print “First root = ” , Root1

Print “Second root = ” , Root2

End

ELSE

Begin

Print “Roots are imaginary”

part\_r = -B / (2 \* A)

part\_i = √-D / (2 \* A)

Root1 = part\_r + i part\_i

Root2 = part\_r - i part\_i

Print “First root = ” , Root1

Print “Second root = ” , Root2

End

End

1. Stop

**PROGRAM**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

float a,b,c,d,rt;

float r1,r2,part\_r,part\_i;

clrscr();//Clears the screen

printf("\n\t\tQuadratic equation\n");

printf("\t\t------------------\n");

printf("\nEnter Coefficients of quadratic eq ax^2+bx+c :\n ");

printf("\nEnter a : ");

scanf("%f",&a);

printf("\nEnter b : ");

scanf("%f",&b);

printf("\nEnter c : ");

scanf("%f",&c);

//Calculates discriminant

d=b\*b-(4\*a\*c);

//Checks for real, equal and imaginary roots

if(d==0)

{ printf("\nRoots are real & equal \n");

r1=-b/(2\*a);

r2=r1;

printf("\nRoot1 = %f \nRoot2 = %f\n",r1,r2); }

else { if(d>0) { printf("\nRoots are real & distinct\n");

rt=sqrt(d);

r1=(-b+rt)/(2\*a);

r2=(-b-rt)/(2\*a);

printf("\nRoot1 = %f \nRoot2 = %f\n",r1,r2);

} else {

printf("\nRoots are complex\n");

part\_r=-b/(2\*a);

part\_i=sqrt(fabs(d));

printf("\nRoot1 = %.2f + i %.2f \n",part\_r,part\_i);

printf("\nRoot2 = %.2f - i %.2f \n",part\_r,part\_i);

} } getch();

}//End of the program

**OUTPUT**

Quadratic equation

------------------------

CASE1 : Roots are real and equal

----------

Enter Coefficients of quadratic eq ax^2+bx+c :

Enter a : 2

Enter b : 4

Enter c : 2

Roots are real & equal

Root1 = -1.000000

Root2 = -1.000000

CASE2: Roots are real and distinct

Quadratic equation

-------------------------

Enter Coefficients of quadratic eq ax^2+bx+c :

Enter a : 2

Enter b : 5

Enter c : 2

Roots are real & distinct

Root1 = -0.500000

Root2 = -2.000000

CASE3 : Roots are complex

---------

**EXPERIMENT NO : 4**

**PRIME NUMBER CHECKING**

**AIM** : Program to check whether the given integer number is prime or not

**ALGORITHM**

1. Start
2. Read the input number to check for prime, NUM
3. IF NUM = 1 THEN

Print “1 is neither prime nor composite”

ELSE

Begin

Repeat for I = 2,3,4,………………………………., NUM/2

Begin

IF NUM % I = 0 THEN

BREAK

End

IF I > NUM /2 THEN

Print NUM , “is a prime number”

ELSE

Print NUM , “is not a prime number”

End

1. Stop

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,num;

clrscr(); //Clears the screen

printf("\n\t\tPRIME NUMBER CHECKING");

printf("\n\t\t---------------------\n");

printf("\nEnter the integer number to check for prime :");

scanf("%d",&num);

if(num==1)

{ printf("\n 1 is neither prime nor composite"); }

else {

/\* Initializing i=2(smallest factor other than one) and

repeating for all values of i less than or equal to

n/2 \*/

for(i=2;i<=num/2;i++)

{ if(num%i==0)

break; }

if( i > num/2 )

printf("\n%d is a prime number",num);

else

printf("\n%d is not a prime number",num); }

getch();

}

**OUTPUT**

CASE1:

----------

PRIME NUMBER CHECKING

---------------------------------------

Enter the integer number to check for prime :15

15 is not a prime number

CASE2:

-----------

PRIME NUMBER CHECKING

----------------------------------------

Enter the integer number to check for prime :2

2 is a prime number

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 5**

**Armstrong**

**AIM :** Program to check whether a given number is armstrong or not

**ALGORITHM**

1. Start
2. Read the input number to check whether Armstrong or not
3. Assign number to num
4. Begin

Repeat if n>0

r←n%10

s←s+(r\*r\*r)

n←n/10

End

1. If s==num

Print “Number is Armstrong”

Else

Print “number not Armstrong”

1. Stop

**Program**

#include<stdio.h>

int main()

{

int x,r,n,i,s=0;

printf("Enter the number to check whether armstrong or not : ");

scanf("%d",&n);

x=n;

while(n>0)

{

r=n%10;

s=s+(r\*r\*r);

n=n/10;

}

if(x==s)

printf("%d is armstrong",x);

else

printf("%d is not armstrong",x);

return 0;

}

**Output**

Enter the number to check whether armstrong or not :

153

153 is armstrong

Enter the number to check whether armstrong or not :

145

145 is not Armstrong

**EXPERIMENT NO : 6**

**sum of ‘n’ natural numbers**

**AIM :** Program to calculate the sum of n natural numbers

**ALGORITHM**

1. Start
2. Read the limit of numbers
3. i←0
4. Enter the numbers

Begin

Repeat if n>i

Read number into num

s←s+num

Increment i

End

1. Print “Sum of n numbers is s”
2. Stop

**Program**

#include<stdio.h>

#define max 50

int main()

{

int i=0,n,s=0,a[max];

printf("Enter the number of natural numbers to find the sum : ");

scanf("%d",&n);

printf("\nEnter numbers\n");

while(i<n)

{

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

s=s+a[i];

i++;

}

printf("\n\nSum of the numbers entered is %d",s);

return 0;

}

**Output**

Enter the number of natural numbers to find the sum : 5

Enter numbers

1

2

3

4

5

Sum of the numbers entered is 15

**EXPERIMENT NO : 7**

**palindrome**

**AIM :** Program to check whether a given number is palindrome or not

**ALGORITHM**

1. Start
2. Read the input number to check whether palindrome or not
3. Assign number to num
4. Begin

Repeat if n>0

r←n%10

s←s\*10+r

n←n/10

End

1. If s==num

Print “Number is Palindrome”

Else

Print “Number is not Palindrome”

1. Stop

**Program**

#include<stdio.h>

int main()

{

int x,r,n,s=0;

printf("Enter the number to check whether a palindrome or not : ");

scanf("%d",&n);

x=n;

while(n>0)

{

r=n%10;

s=s\*10+r;

n=n/10;

}

if(s==x)

printf("%d is a palindrome number",x);

else

printf("%d is not a palindrome number",x);

return 0;

}

**Output**

Enter the number to check whether a palindrome or not :

121

121 is a palindrome number

Enter the number to check whether a palindrome or not :

123

123 is not a palindrome number

**EXPERIMENT NO : 8**

**Fibonacci series**

**AIM :** Program to calculate the Fibonacci series upto ‘n’

**ALGORITHM**

1. Start
2. Read the input number as limit
3. Assign x=0,y=1 and s=0
4. Display Fibonnaci Series
5. Print x and y
6. Begin

Repeat if s<=n

s←x+y

Print s

x←y

y←s

End

1. Stop

**Program**

#include<stdio.h>

int main()

{

int x,y,s=0,n;

printf("Enter the range : ");

scanf("%d",&n);

x=0;y=1;

printf("\n\t\tFibonacci series\n%d\n%d\n",x,y);

while(s<=n)

{

s=x+y;

if(s<=n)

printf("%d\n",s);

x=y;

y=s;

}

return 0;

}

**Output**

Enter the range : 20

Fibonacci series

0

1

1

2

3

5

8

13

**EXPERIMENT NO : 9**

**binary to Decimal**

**AIM :** Program to convert a binary to decimal number .

**ALGORITHM**

1. Start
2. Declare an integer array a
3. Read the input number n, to be converted into binary
4. i←0
5. Begin

Repeat until n>0

Assign integer array a with n%2

n←n/2;

Increment loop counter i

End

1. Display elements in array a in descending order
2. Stop

**Program**

#include<stdio.h>

int main()

{

int n,i,a[20],j,c;

printf("Enter the decimal number to convert to binary : ");

scanf("%d",&n);

printf("Binary equivalent of %d is ",n);

i=0;

while(n>0)

{

a[i++]=n%2;

n=n/2;

}

for(j=i-1;j>=0;j--)

printf("%d",a[j]);

return 0;

}

**Output**

Enter the decimal number to convert to binary : 12

Binary equivalent of 12 is 1100

Enter the decimal number to convert to binary : 32

Binary equivalent of 32 is 100000

**EXPERIMENT NO : 10**

**Trace and norm of a matrix**

**AIM :** Program to calculate trace and norm of a matrix

**ALGORITHM**

1. Start
2. Declare integer arrays a,b and d
3. Decalre variables r,c as number of rows and columns respectively
4. Declare trace,trace1,sum\_a,sum\_b
5. Initialize loop counter i and j as 0
6. Read the order of matrices a and b
7. Read values into matrix a using for loop
8. Begin

Repeat until i<r and j<c

sum\_a←sum\_a+a[i][j]\*a[i][j];

End

1. Read values into matrix b using for loop
2. Begin

Repeat until i<r and j<c

sum\_b←sum\_b+b[i][j]\*b[i][j];

End

1. Display matrix a and b
2. Add the matrices

Begin

Repeat while i<r and j<c

d[i][j]←a[i][j]+b[i][j]

Increment loop counter i and j

End

1. Display array d
2. Display norm of matrix a sqrt(sum\_a)
3. Display norm of matrix a sqrt(sum\_b)
4. If r==c calculate trace of both matrix a and b
5. Trace of matrix a
6. Begin

Repeat until i<r and j<c

if i==j

trace←trace+a[i][j]

1. Display trace of matrix a
2. Trace of matrix b
3. Begin

Repeat until i<r and j<c

if i==j

trace←trace+b[i][j]

1. End

**Program**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int i,j,a[10][10],b[10][10],d[10][10],r,c,trace=0,trace1=0,flag=0;

float sum\_a=0,sum\_b=0;

clrscr();

printf("Enter the order of matrices A and B : ");

scanf("%d%d",&r,&c);

if(r==c)

flag++;

printf("\nEnter elements of matrix A \n");

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

{

for(j=0;j<c;j++)

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

}

printf("\nEnter elements of matrix B \n");

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

{

for(j=0;j<c;j++)

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

}

printf("\n\t\tDisplaying elements");

printf("\nElements of matrix A\n");

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

{

for(j=0;j<c;j++)

{

printf("%d",a[i][j]);

sum\_a+=a[i][j]\*a[i][j];

}

printf("\n");

}

printf("\nElements of matrix B\n");

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

{

for(j=0;j<c;j++)

{

printf("%d",b[i][j]);

sum\_b+=b[i][j]\*b[i][j];

}

printf("\n");

}

// Adding matrices

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

{

for(j=0;j<c;j++)

d[i][j]=a[i][j]+b[i][j];

}

printf("\nSum of matrix A and B\n");

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

{

for(j=0;j<c;j++)

printf("%d",d[i][j]);

printf("\n");

}

//norm calculation

printf("Norm of matrix A is %f\n",sqrt(sum\_a));

printf("Norm of matrix B is %f\n",sqrt(sum\_b));

//trace calculation

if(flag>0)

{

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

{

for(j=0;j<c;j++)

{

if(i==j)

trace+=a[i][j];

}

}

printf("\nTrace of matrix(sum of diagonals) of matrix A : %d ",trace);

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

{

for(j=0;j<c;j++)

{

if(i==j)

trace1+=b[i][j];

}

}

printf("\nTrace of matrix(sum of diagonals) of matrix B: %d ",trace1);

}

else

printf("\nTrace can be only determined for square matrices\n ");

getch();

}

**Output**

Enter elements of A

1 2 3

4 5 6

7 8 9

Enter elements of B

1 0 0

0 1 0

0 0 1

Displaying elements

Enter elements of A

1 2 3

4 5 6

7 8 9

Enter elements of B

1 0 0

0 1 0

0 0 1

Sum of matrix A and B

2 2 3

4 6 6

7 8 10

Norm of matrix A is 16.881943

Norm of matrix B is 1.732051

Trace of matrix(sum of diagonals) of matrix A : 15

Trace of matrix(sum of diagonals) of matrix B : 3

**EXPERIMENT NO : 11**

**Basic mathematical operations using menu driven program**

**AIM :** Program to perform basic mathematical operations using a menu driven program

**ALGORITHM**

1. Start
2. Read two numbers
3. Read a variable op as operation to be performed on the numbers
4. Display 1.addition 2.subtraction 3.multiplication 4.division
5. Begin

Repeat while ch==’y’ or ch==’Y’

By using switch perform the following operations

If op is 1 a+b

if op is 2 a-b

if op is 3 a\*b

if op is 4 a/b

Display result based on the operation

End

1. Stop

**Program**

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,op;

char ch;

clrscr();

printf("\t\tMenu Driven Program for Basic Mathematical Operations\n");

do

{

printf("Enter values for A and B : \n");

scanf("%d%d",&a,&b);

printf("\n Press 1 for Addition\n Press 2 for Subtraction\n Press 3 for Multiplication\n Press 4 for Division\n");

printf("\nEnter operator : ");

scanf("%d",&op);

switch(op)

{

case 1: printf("\n%d + %d = %d",a,b,a+b);

break;

case 2: printf("\n%d - %d = %d",a,b,a-b);

break;

case 3: printf("\n%d \* %d = %d",a,b,a\*b);

break;

case 4: printf("\n%d / %d = %f",a,b,a/b);

break;

default : printf("\nNot valid entry\n");

}

printf("\nDo you want to continue ? (y/n) : ");

scanf("%c",&ch);

}while(ch=='y'||ch=='Y');

getch();

}

**Output**

Menu Driven Program for Basic Mathematical Operations

Enter values for A and B : 7

3

Press 1 for Addition

Press 2 for Subtraction

Press 3 for Multiplication

Press 4 for Division

Enter operator :3

7 \* 3 = 21.000

Do you want to continue ? (y/n) :y

Enter values for A and B : 8

4

Press 1 for Addition

Press 2 for Subtraction

Press 3 for Multiplication

Press 4 for Division

Enter operator :1

8 + 4 = 12.000

Do you want to continue ? (y/n) :y

Enter values for A and B :6

8

Press 1 for Addition

Press 2 for Subtraction

Press 3 for Multiplication

Press 4 for Division

Enter operator :5

Not valid entry

Do you want to continue ? (y/n) :n

**EXPERIMENT NO : 12**

**AIM:** Write a program to convert a number from binary to decimal.

**ALGORITHM**

1. Start
2. Declare an integer array bi
3. Read the input binary number num, to be converted into decimal
4. p←0
5. Begin

Repeat until num>0

Assign integer array bi with num%10

num=num/10

p=p+1

End loop

1. q=p-1
2. r=0
3. Repeat loop until q>=0
   1. r=r+(bi[q]\*pow(2,q));
   2. q=q-1
4. display the values of r.

End

**PROGRAM**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**int i,j,a[10][10],p,q,l,s;**

**clrscr();**

**printf("Enter the order of matrix:");**

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

**printf("\nEnter elements of matrix A \n");**

**for(i=0;i<p;i++)**

**{**

**for(j=0;j<q;j++)**

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

**}**

**printf("\n\t\tDisplaying elements");**

**printf("\nElements of matrix A\n");**

**for(i=0;i<p;i++)**

**{**

**for(j=0;j<q;j++)**

**{**

**printf("%d",a[i][j]);**

**}**

**printf("\n");**

**}**

**l=0;**

**s=a[0][0];**

**for(i=0;i<p;i++)**

**{**

**for(j=0;j<q;j++)**

**{**

**if(l<a[i][j])**

**l=a[i][j];**

**if(a[i][j]<s)**

**s=a[i][j];**

**}**

**}**

**printf("biggest value: %d",l);**

**printf("lowest value: %d",s);**

**getch();**

**}**

**OUTPUT**

**Enter binary no:1010**

**Decimal: 10**

**EXPERIMENT NO : 13**

**AIM:** Write a program to find the largest and smallest values of a matrix.

**ALGORITHM:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int i,j,a[10][10],p,q,l,s;

clrscr();

printf("Enter the order of matrix:");

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

printf("\nEnter elements of matrix A \n");

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

{

for(j=0;j<q;j++)

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

}

printf("\n\t\tDisplaying elements");

printf("\nElements of matrix A\n");

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

{

for(j=0;j<q;j++)

{

printf("%d",a[i][j]);

}

printf("\n");

}

l=0;

s=a[0][0];

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

{

for(j=0;j<q;j++)

{

if(l<a[i][j])

l=a[i][j];

if(a[i][j]<s)

s=a[i][j];

}

}

printf("biggest value: %d",l);

printf("lowest value: %d",s);

getch();

}

**OUTPUT**

Enter the order of matrix: 2 2

Enter elements of matrix A

1 2 3 4

Displaying elements

Elements of matrix A

1 2

3 4

Biggest value:4

Lowet value:1

**Cycle-2**

**EXPERIMENT NO : 1**

**Vowels in a string**

**AIM :** Program to calculate the number of vowels in a given string

**ALGORITHM**

1. Start
2. Declare character array c and character variable ch
3. Initialize loop counter k and a,e,i,o,u vowel counters to 0
4. Enter the string into a character array
5. ch←c[k]
6. Begin

Repeat until ch!=’\0’

Read character from character array one by one

Using switch increment counters a,e,i,o,u on encounter of a,e,i,o,u in the given string

Increment loop counter k

End

1. Display counter a,e,i,o,u to display number of a’s e’s i’s o’s and u’s in the given string
2. Stop

**Program**

#include<stdio.h>

int main( )

{

char ch,c[50];

int k=0,a=0,e=0,i=0,o=0,u=0;

printf("\nEnter the string to find the number of vowels : \n");

scanf("%s",c);

ch=c[k];

while(ch!='\0')

{

switch(ch)

{

case 'a' : a++;

break;

case 'e' : e++;

break;

case 'i' : i++;

break;

case 'o' : o++;

break;

case 'u' : u++;

break;

default: printf(“\nNo vowels in string”);

}

ch=c[++k];

}

printf("\nNumber of vowel a is %d\n",a);

printf("\nNumber of vowel e is %d\n",e);

printf("\nNumber of vowel i is %d\n",i);

printf("\nNumber of vowel o is %d\n",o);

printf("\nNumber of vowel u is %d\n",u);

return 0;

}

**Output**

Enter the string to find the number of vowels :

dictionaryNumber of vowel a is 1Number of vowel e is 0Number of vowel i is 2

Number of vowel o is 1Number of vowel u is 0

Enter the string to find the number of vowels : beautifulNumber of vowel a is 1Number of vowel e is 1Number of vowel i is 1Number of vowel o is 0Number of vowel u is 2

**EXPERIMENT NO : 2**

**Print triangle of asterisks**

**AIM :** Program to print the given format of n number of rows

**\***

**\* \***

**\* \* \***

**ALGORITHM**

1. Start
2. Enter number of rows to print
3. Begin

Repeat until

End

1. Stop

**Program**

#include<stdio.h>

#include<conio.h>

void main()

{

int p,q,r,x,k;

clrscr();

q=0;

printf("Number of rows :");

scanf("%d",&r);

printf("\nPascal's Triangle:\n");

while(q<r)

{

for(p=40-3\*q;p>0;--p)

printf(" ");

for(x=0;x<=q;x++)

{ k=0;

printf("%c",'\*');

while(k<5)

{

printf(" ");

k++;

}

}

printf("\n");

q++;

}

getch();

}

**Output**

Number of rows : 4

Pascal’s Triangle:

\*

\* \*

\* \* \*

\* \* \* \*

**EXPERIMENT NO : 3**

**SIne and cosine series**

**AIM** : Program to find the sum of the sine and cosine series:

Sine : X – X3 + X5 - X7 + ……

3! 5! 7!

Cosine : 1 - X2 + X4 - X6 + ……

2! 4! 6!

**ALGORITHM**

1. Start
2. Read the limit of the series
3. Read the value of x in degrees
4. tmp ← x
5. x ← (x \* 3.14) / 180
6. sin\_term ← x
7. cos\_term ← x
8. Repeat for I = 1,2,3,……………………..n

Begin

sin\_sum ← sin\_sum + sin\_term

sin\_term ← sin\_term \* (-x2 / 2i(2i+1))

cos\_sum ← cos \_sum + cos \_term

cos \_term ← cos \_term \* (-x2 / 2i(2i-1))

End

1. Stop

**Program**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int i,n;

float sin\_sum=0,cos\_sum=0,x,term,tmp,sin\_term,cos\_term;

clrscr();

printf("\n\n\t\tSUM OF SINE AND COSINE SERIES\n");

printf("\n\nEnter the limit of the series : ");

scanf("%d",&n);

printf("\n\nEnter the value of x in degrees : ");

scanf("%f",&x);

tmp = x;

//Converts x into radians

x= (x \* 3.14) / 180;

//First term of sine and cosine series

sin\_term = x;

cos\_term = 1;

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

{

sin\_sum += sin\_term;

sin\_term \*= - x\*x / ( 2\*i \* (2\*i+1) );

cos\_sum += cos\_term;

cos\_term \*= - x\*x / ( 2\*i \* (2\*i-1) );

}

printf("\n\nSine(%f) = %f",tmp,sin\_sum);

printf("\n\n\nCosine(%f) = %f",tmp,cos\_sum);

getch();

}

**Output**

SUM OF SINE AND COSINE SERIES

Enter the limit of the series : 30

Enter the value of x in degrees : 90

Sine(90.000000) = 1.000000

Cosine(90.000000) = 0.000796

**EXPERIMENT NO : 4**

**factorial of a number using recursion**

**AIM** : Program to calculate factorial of a number using recursion

**ALGORITHM**

1. Start
2. Read the number to calculate the factorial to n
3. Call function fact(n)
4. Function fact(x)
5. Begin

Repeat until x>0

If(n=1 or n=0) then

Return n

Else

x←x\*fact(x-1)

Return x

End

1. Stop

**Program**

#include<stdio.h>

int main()

{

int n;

clrscr();

printf("Enter value for n: ");

scanf("%d",&n);

printf("\n\nFactorial of %d : %d",n,fact(n));

return 0;

}

int fact(int x)

{

if(x==1)

return x;

else

x=x\*fact(x-1);

return x;

}

**Output**

Enter value for n: 3

Factorial of 3 : 6

Enter value for n: 4

Factorial of 3 : 24

**EXPERIMENT NO : 5**

**substring of a given string**

**AIM** : Program to find the substring of a given string:

**ALGORITHM**

1. Start
2. Read main string and a sub string
3. Read the value of x in degrees
4. Use strstr function to see whether the substring is present in the main string
5. If ch←1
6. Print substring found
7. Else
8. Substring not found
9. Stop

**Program**

#include <string.h>  
#include <stdio.h>   
intmain()

{  
  char \*ch;  
 char str1[15],str2[10];

printf(“Enter main string:\n”);

scanf(“%s”,str1);

printf(“Enter substring:\n”);

scanf(“%s”,str2);  
  ch = strstr(str1,str2);

if(ch==1)  
   printf(“\nSubstring found”);

else

Printf(“\nSubstring not found”);

getch();

}

**Output**

Enter main strings

Where there is a will, there is away

Enter sub string

will

Substring found

Enter main strings

Where there is a will, there is away

Enter sub string

many

Substring not found

**EXPERIMENT NO : 6**

**concatenate two strings without using string functions**

**AIM** : Program to perform concatenation of two strings without using built in functions

**ALGORITHM**

1. Start
2. Read two strings into arrays str1 and str2
3. Initialize loop counters i,j counters c and c1 to 0
4. Read characters one by one from individual arrays and increment counter c and c1
5. Begin

Repeat until c1>i

str1[c]←str2[i]

Append \0 to str1

End

1. Print string str1
2. Stop

**Program**

#include<stdio.h>

int main()

{

char str1[30],str2[30];

int i=0,j=0,c=0,c1=0;

printf("Enter two strings \n");

scanf("%s%s",str1,str2);

while(str1[i]!='\0')

{

c++;

i++;

}

while(str2[j]!='\0')

{

c1++;

j++;

}

for(i=0;i<c1;i++)

{

str1[c]=str2[i];

c++;

}

str1[c]='\0';

printf("%s",str1);

return 0;

}

**Output**

Enter two strings

Hai

Helo

HaiHelo

**EXPERIMENT NO : 7**

**ADDITION & MULTIPLICATION OF TWO COMPLEX NUMBERS**

**AIM** : Program to perform addition and mulitiplication of two complex numbers.

**ALGORITHM**

1. Start
2. Read 1st complex no’s coefficients to a and b
3. Read 2nd complex no’s coefficients to c and d
4. Add a+c and b+d and form addition result.
5. X=ac-ad;
6. Y=bc+ad;

Print X+iY.

1. Stop

**PROGRAM**

Program

#include<stdio.h>

int main()

{

int a,b,c,d,x,y;

printf("\nEnter the first complex number:");

scanf("%d%d",&a,&b);

printf("\nEnter the second complex number:");

scanf("%d%d",&c,&d);

if(b<0)

printf("%d-i%d\n",a,-b);

else

printf("%d+i%d\n",a,+b);

if(d<0)

printf("%d-i%d\n",c,-d);

else

printf("%d+i%d\n",c,+d);

printf("\nADDITION ");

x=a+c;

y=b+d;

if(y>0)

printf("%d-i%d",x,y);

else

printf("%d+i%d",x,y);

printf("\n\nMULTIPLICATION ");

x=a\*c-b\*d;

y=b\*c+a\*d;

if(y>0)

printf("%d-i%d",x,y);

else

printf("%d+i%d",x,y);

return 0;

}

Output

Enter the first complex number:

1 2

Enter the second complex number:

1 -3

1+i2

1-i3

ADDITION 2+i-1

MULTIPLICATION 7+i-1

**8.Write a program to accept the employee details and calculate each of the employee commission. Given commission is 20% of the salary.**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

struct employee

{

int empno,salary;

char name[15],sex[10];

float comm;

}emp[10];

int lim,i=0;

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

scanf("%d",&lim);

while(i<lim)

{

printf("Enter the employee details: ");

printf("\nEmployee number :");

scanf("%d",&emp[i].empno);

printf("\nEmployee name : ");

scanf("%s",emp[i].name);

printf("\nSex : ");

scanf("%s",emp[i].sex);

printf("\nSalary : ");

scanf("%d",&emp[i].salary);

i++;

}

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

{

emp[i].comm=emp[i].salary\*.2;

printf("\nCommission of employee %d is %f",i+1,emp[i].comm);

}

getch();

}

Output

Enter the number of employees :2

Enter the employee details:

Employee number :1

Employee name :anu

Sex :female

Salary :30000

Enter the employee details:

Employee number :2

Employee name :manu

Sex :male

Salary :25000

Commission of employee 1 is 6000.000000

Commission of employee 2 is 5000.000000

**9. Write a program to print the student record. Accept name, register nos. and marks in three subjects for each student and compute the class average of each subject**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

int i=0,lim,p=0,c=0,m=0;

float avg1,avg2,avg3;

struct student

{

char name[10];

int regno,phy,chem,maths;

}s[10];

clrscr();

printf("\nEnter the number of students whos details to be entered : ");

scanf("%d",&lim);

while(i<lim)

{

printf("Enter student name : ");

scanf("%s",s[i].name);

printf("\nEnter regno : ");

scanf("%d",&s[i].regno);

printf("\nEnter the marks for physics : ");

scanf("%d",&s[i].phy);

printf("\nEnter the marks for chemistry : ");

scanf("%d",&s[i].chem);

printf("\nEnter the marks for maths : ");

scanf("%d",&s[i].maths);

i++;

}

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

{

p=p+s[i].phy;

c=c+s[i].chem;

m=m+s[i].maths;

}

avg1=p/lim;

avg2=c/lim;

avg3=m/lim;

printf("\nClass average of physics : %.3f",avg1);

printf("\nClass average of chemistry : %.3f",avg2);

printf("\nClass average of maths : %.3f",avg3);

getch();

}

Output

Enter the number of students whos details to be entered :5

Enter student name : Mini

Enter regno : 25

Enter the marks for physics : 47

Enter the marks for chemistry : 39

Enter the marks for maths : 42

Enter student name : Akhil

Enter regno : 33

Enter the marks for physics : 40

Enter the marks for chemistry : 35

Enter the marks for maths : 41

Enter student name : Mathew

Enter regno : 20

Enter the marks for physics : 33

Enter the marks for chemistry : 39

Enter the marks for maths : 41

Enter student name : Manu

Enter regno : 29

Enter the marks for physics : 47

Enter the marks for chemistry : 42

Enter the marks for maths : 42

Enter student name : Smith

Enter regno : 19

Enter the marks for physics : 48

Enter the marks for chemistry : 32

Enter the marks for maths : 42

Class average of physics :43.000

Class average of chemistry :37.400

Class average of maths :41.600

**10.Write a program to search an element in an array using binary search method.**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

int i,j,x,n,t;

int low,high,mid,a[20];

clrscr();

printf("Enter the n value:");

scanf("%d",&n);

printf("\nEnter the numbers");

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

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

//sorting elements....

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;

}

}

}

// printf("\nSorted array\n");

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

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

printf("\nEnter the search element:");

scanf("%d",&x);

low=0;

high=n;

while(low<=high)

{

mid=(low+high)/2;

if(x<a[mid])

{

high=mid-1;

}

else if(x>a[mid])

{

low=mid+1;

}

else if(x==a[mid])

{

printf("\n Number obtained at position %d",mid+1);

break;

}

else

printf(“\n Number not found”);

}

getch();

}

Output

Enter the n value:4

Enter the numbers 56 78 23 45

Sorted array 23 45 56 78

Enter the search element:78

Number obtained at position 4

Enter the n value:3

Enter the numbers 44 70 21

Sorted array 21 44 70

Enter the search element:87

Number not found

***Cycle-3***

**1. Write a program to create a file to store details of n students – A file named student.dat contain information such as rollno, name, and total marks**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

FILE \*fp;

int i=0,lim;

struct student

{

char name[10];

int rollno,phy,chem,maths;

float tot\_marks;

}s[10];

clrscr();

fp=fopen("student.dat","w");

printf("\nEnter the number of students whos details to be entered : ");

scanf("%d",&lim);

printf("\nEnter the following details : Name of student,Rollno of student,Marks of subjects \n");

while(i<lim)

{

printf("\nEnter student name : ");

scanf("%s",s[i].name);

printf("\nEnter rollno : ");

scanf("%d",&s[i].rollno);

printf("\nEnter the marks for physics : ");

scanf("%d",&s[i].phy);

printf("\nEnter the marks for chemistry : ");

scanf("%d",&s[i].chem);

printf("\nEnter the marks for maths : ");

scanf("%d",&s[i].maths);

s[i].tot\_marks=s[i].phy+s[i].chem+s[i].maths;

printf("\nTotal marks : %f\n",s[i].tot\_marks);

i++;

}

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

fprintf(fp,"%s %d %f\n",s[i].name,s[i].rollno,s[i].tot\_marks);

getch();

}

Output

Enter the number of students whos details to be entered 3

Enter the following details : Name of student,Rollno of student,Marks of subjects

Enter student name :Anu

Enter rollno :101

Enter the marks for physics :46

Enter the marks for chemistry :47

Enter the marks for maths :49

Total marks : 142.000000

Enter student name :Veena

Enter rollno :102

Enter the marks for physics :39

Enter the marks for chemistry :45

Enter the marks for maths :42

Total marks : 126.000000

Enter student name :Vivek

Enter rollno :103

Enter the marks for physics :46

Enter the marks for chemistry :34

Enter the marks for maths :49

Total marks : 129.000000

**Student.dat**

NAME :Anu

ROLLNO :101

TOTAL MARKS :142.000000

NAME :Veena

ROLLNO :102

TOTAL MARKS :126.000000

NAME :Vivek

ROLLNO :103

TOTAL MARKS :129.000000

**2. Write a program to merge two files**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

FILE \*fp,\*fp1;

char ch;

clrscr();

//opening first file in read mode

fp=fopen("first.txt","r");

fp1=fopen("mainfile.txt","w");

printf("First file content : \n");

ch=getc(fp);

printf("%c",ch);

while(ch!=EOF)

{

putc(ch,fp1);

ch=getc(fp);

printf("%c",ch);

}

fclose(fp);

fp=fopen("second.txt","r");

printf("\nSecond file content : \n");

ch=getc(fp);

printf("%c",ch);

while(ch!=EOF)

{

putc(ch,fp1);

ch=getc(fp);

printf("%c",ch);

}

fclose(fp);

fclose(fp1);

printf("\nMerged file with contents of both files\n");

fp1=fopen("mainfile.txt","r");

while((ch=getc(fp1))!=EOF)

printf("%c",ch);

getch();

}

Output

First file content : helo everybody.

Second file content : god bless you.Thanku.

Merged file with contents of both files

helo everybody.god bless you.Thanku.

**3. Write a program to apply linear search to a set of n numbers by using function**

Program

#include<stdio.h>

#include<conio.h>

int i,n,c;

void main()

{

int a[10],m;

c=0;

printf("\nEnter the size of an array");

scanf("%d",&n);

printf("\nEnter the elements of the array");

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

{

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

}

printf("\nThe elements of an array are");

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

{

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

}

printf("\nEnter the number to search");

scanf("%d",&m);

c=lin\_search(m,a);

if(c>0)

printf("\nThe number is found");

else

printf("\nThe number is not in list");

getch();

}

int lin\_search(int m,int a[])

{

int x;

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

{

if(a[i]==m)

{

x=1;

break;

}

}

return x;

}

Output

Enter the size of an array 5

Enter the elements of the array 49 58 23 37 10

The elements of an array are 49 58 23 37 10

Enter the number to search 23

The number is found

Enter the size of an array 5

Enter the elements of the array 34 56 12 89 90

The elements of an array are 34 56 12 89 90

Enter the number to search 91

The number is not in list

**4. Write a program to find the largest and smallest element in an array using pointers**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

int a[20],i,lim;

int \*p,\*l,\*s;

clrscr();

printf("Enter limit :");

scanf("%d",&lim);

printf("\nEnter values : ");

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

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

p=a;

\*l=0;

\*s=a[0];

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

{

if(\*l<\*p)

\*l=\*p;

if(\*s>\*p)

\*s=\*p;

p++;

}

printf("\nLargest element : %d",\*l);

printf("\nSmallest element : %d",\*s);

getch();

}

Output

Enter limit :5

Enter values :32

67

12

90

06

Largest element : 90

Smallest element : 6

**5. Write a program to read a file and find the number of sentences, words, and vowels**

Program

#include<stdio.h>

#include<conio.h>

void main()

{

FILE \* fp;

char ch;

int w=0,v=0,s=0;

clrscr();

fp=fopen("helo.txt","r");

printf("T E X T : ");

ch=getc(fp);

while(ch!=EOF)

{

printf("%c",ch);

if(ch=='.')

{s++;w++;}

else if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch==’A’||ch==’E’||ch==’I’||ch==’O’||ch==’U’)

v++;

else if(ch==' ')

w++;

ch=getc(fp);

}

printf("\nThe total number of sentences is %d words is %d and vowels is %d",s,w,v);

getch();

}

Output

T E X T : Hai.helo everybody.U need to work hard.God bless you.Thankyou.

The total number of sentences is 5 words is 12 and vowels is 19

**CHECK A GIVEN YEAR IS A LEAP YEAR OR NOT.**

**AIM** : Program to find out whether a given year is a leap year or not.

**INPUT SPECIFICATION :** year - An year to check whether it’s a leap year

**OUTPUT SPECIFICATION :** Printing whether the year is a leap year or not.

**ALGORITHM**

1. Start
2. Reading a year, year
3. Checking whether year%4 equal to 0 and year%100!=0 or year%400 is equal to 0
4. Then given year is a leap year and go to 6.
5. Otherwise the given year is not a leap year.
6. Stop.

**PROGRAM**  **DESCRIPTION**

The year is said to be a leap year if it is divisible by 4 and not divisible by 100 or divisible by 400. Using this idea we have check whether the given year is a leap year or not.

**PROGRAM**

#include<stdio.h>

int main()

{

int year;

printf(" Enter a year\n");

scanf(" %d",&year);

if((year%4==0 && year%100!=0)|| (year%400)==0)

printf("Leap year\n");

else

printf("Not a leap year\n");

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc leapyear.c

meera@meera-laptop:~$ ./a.out

Enter a year

1996

Leap year

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 3**

**SUM OF DIGITS AND REVERSING THE NUMBER**

**AIM** : Program to find out the sum of all digits of a given number and then reverse the number

**INPUT SPECIFICATION :** num - number to reverse

**OUTPUT SPECIFICATION :** printing the reversed number and sum of the digits.

**ALGORITHM**

1. Start
2. Read the input number, num
3. Assign sum as 0
4. Repeat the steps till num not equal to 0

4.1) Divide the number with 10 and take its reminder and print the digit.

4.2) Add the reminder that is the last digit to sum

4.3) Divide the number by 10

5. Print the sum

1. Stop

**PROGRAM DESCRIPTION**

A number n is taken and let a variable called sum is used

to store the determined sum. The digit in the position is

extracted using the modulo 10 division and added to the

sum and print the digit, After extracting the unit’s digit,

the number gets reduced. The above process is extracting

the unit’s digit and summing up is repeated till the number

becomes zero.

**PROGRAM**

#include<stdio.h>

int main()

{

int num,rev,sum=0,digit;

printf("Enter a number\n");

scanf("%d",&num);

printf(" Number before reversing %d\n",num);

printf(" Reversed number=");

while(num!=0)

{

digit=num%10;

sum=sum+digit;

num=num/10;

printf("%d",digit);

}

printf("\nSum of digits=%d\n", sum);

}

**OUTPUT**

meera@meera-laptop:~$ cc sum\_reverse.c

meera@meera-laptop:~$ ./a.out

Enter a number

345

Number before reversing 345

Reversed number=543

Sum of digits=12

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 4**

**PRIME NUMBER CHECKING**

**AIM** : Program to generate prime numbers between 1 to n:

**INPUT SPECIFICATION :** num - number to check for prime

**OUTPUT SPECIFICATION :** j- prime number series.

**ALGORITHM**

1. Start
2. Read the range as n
3. Begin
4. Repeat for J=2, 3,4,…………………………………………n
5. Repeat for I = 2,3,4,………………………………., n/2

Begin

IF J % I = 0 THEN

End

1. IF I >J /2 THEN

Print j

End

1. Stop

**PROGRAM DESCRIPTION**

A number n is said to be prime if and only if it is

divisible by 1 and itself and not if it is divisible

by any number in the range 2 to (n-1).

The procedure is to check for the divisibility of the

given number n by all numbers up to n/2.This is because

a number n is never divisible by a number greater than n/2.

Testing for divisibility is done repeatedly under the

control of a counted loop structure.Here the program

control may come out of the loop normally after completing

all the iterations or abnormally after a successful

testing for division.

When the control comes out normally ,the loop control

variable will be greater than (n/2) and the the given

integer will be a prime number (as it was not divisible

by any numbers in the range 2 to (n/2)) rather 2 to (n-1)

When the control comes out abnormally , the loop control

variable will be less than or equal to (n/2) and the given

integer will not be a prime number and exit from the loop.

**PROGRAM**

#include<stdio.h>

int main()

{

int j,i,n;

printf(" Enter the value of n\n");

scanf("%d",&n);

printf(" Prime numbers are:-\n");

for(j=2;j<=n;j++)

{

for(i=2;i<=j/2;i++)

if(j%i==0)

break;

if(i>j/2)

{

printf("%d\t",j);

}

}

printf("\n");

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc prime.c

meera@meera-laptop:~$ ./a.out

Enter the value of n

12

Prime numbers are:-

2 3 5 7 11

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 5**

**FIBNOCCI SERIES**

**AIM** : Program to generate fibnocci series between 1 to n.

**INPUT SPECIFICATION :** range - The number of elements

**OUTPUT SPECIFICATION :** Print the fibnocci series upto the range,c.

**ALGORITHM**

1. Start
2. Read the range, range.
3. Assign a←0 and b ← 1
4. Print a and b.
5. Repeat for I =0,1,2,3,…………….range
6. Begin
7. c← a+b
8. print c
9. a←b
10. b←c
11. End
12. Stop

**PROGRAM DESCRIPTION**

‘a’ and ‘b’ be the first and second terms to bigin with.

The third term obtained with adding a and b.

For generating the fourth term, made a as b and b as c.

This procedure is repeated n times for generating the n terms.

**PROGRAM**

#include<stdio.h>

int main()

{

int i,range,a=0,b=1,c;

printf(" Enter the number range of fibnacci series\n");

scanf("%d",&range);

printf("Fibnocci series of range %d are\n",range);

printf(" %d %d ",a,b);

for(i=3;i<=range;i++)

{

c=a+b;

printf("%d ",c);

a=b;

b=c;

}

printf("\n\n");

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc fibnacci.c

meera@meera-laptop:~$ ./a.out

Enter the number range of fibnacci series

10

Fibnocci series of range 10 are

0 1 1 2 3 5 8 13 21 34

---------------------------------------------------------------------------------------------------------------------\*/

**CYCLE-11**

**EXPERIMENT NO : 6**

**ARMSTRONG NUMBER SERIES**

**AIM** : Program to generate armstrong series between 1 to n.

**INPUT SPECIFICATION :** na - The number of elements

**OUTPUT SPECIFICATION :** Print the armstrong series upto the range,cube.

**ALGORITHM**

1. Start
2. Read the range, range.
3. Repeat for I =1,2,3,…………….range
4. Begin
5. cube←0
6. n←i
7. Repeat till n !=0
8. Begin
9. r←n%10
10. cube←cube+ r\*r\*r
11. n←n/10
12. End
13. Check whether cube is equal to i
14. Print i
15. End
16. Stop

**PROGRAM DESCRIPTION**

A number range is taken as the upper limit of the Armstrong number series.

Now from 1 to till range we have to check the sum of cubes of the digit is

equal to that number, the digit in the position is extracted using the modulo

10 division and finding the cube of that digit and add to sum, After extracting

the unit’s digit, the number gets reduced. The above process is extracting

the unit’s digit and cubing the digit and adding repeats till the number become

zero. Now we have to check the summed cube and the number is same. If its same

printas it is the Armstrong number, otherwise don’t print.

**PROGRAM**

#include<stdio.h>

int main()

{

int na,i,cube,r,n;

printf("\n Enter the upper limit of the armstrong number\n");

scanf("%d",&na);

printf("Armsrong numbers of range 1 to %d are\n",na);

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

{

cube=0;

n=i;

while(n!=0)

{

r=n%10;

cube=cube+(r\*r\*r);

n=n/10;

}

if(cube==i)

{

printf("%d\t",i);

}

}

printf("\n");

return 0;

}

**OUTPUT**

student@user-desktop:~$ cc armstrong.c

student@user-desktop:~$ ./a.out Enter the upper limit of the armstrong number

500

Armsrong numbers of range 1 to 500 are

1 153 370 371 407

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 7**

**simple calculator**

**AIM** : A menu driven program of a simple calculator

**INPUT SPECIFICATION :** a and b- Two numbers to be added/subtracted /multiply/division/modulus

choice**-** character specify which action to be performed.

**OUTPUT SPECIFICATION :** Print the sum/difference/product/remainder according to the choice of the user.

**ALGORITHM**

1. Start
2. Read two numbers, a and b.
3. Read a character, choice.
4. Print the Menu.
5. If choice is equal to ‘A’ or ‘a’, then add the two numbers and print the sum.
6. If choice equal to ‘S’ or ‘s’, then subtract two numbers and print the difference
7. If choice equal to ‘M’ or ‘m’, then multiply two numbers and print the product.
8. If choice is equal to ‘D’ or ‘d’ then divide two numbers and print the result.
9. Else print Wrong choice.
10. Stop

**program DESCRIPTION**

Program to print the menu and according to the users choice perform

the appropriate result. In thios program if the user choose ‘A’ or ‘a’

then add two numbers. If the user choose ‘S’ or ‘s’ then subtract two

numbers, if the user choose ‘M’ or ‘m’ then multiply two numbers,

if the user choose ‘D’ or ‘d’ then divide two numbers.

**program**

#include<stdio.h>

int main()

{

float a,b,c;

char choice;

printf("\t\t MENU\n\n");

printf("\t\tA:Addition\n\n");

printf("\t\tS:Subtration\n\n");

printf("\t\tM:Multiplication\n\n");

printf("\t\tD:Division\n\n");

printf("Enter your choice\n");

choice=getchar();

printf("\nEnter two numbers\n");

scanf("%f%f",&a,&b);

printf("\n\n");

switch(choice)

{

case 'A':

case 'a': c=a+b;

printf("The sum of %f and %f is %f\n",a,b,c);

break;

case 'S':

case 's': c=a-b;

printf("The difference of %f and %f is %f\n",a,b,c);

break;

case 'M':

case 'm': c=a\*b;

printf("The product of %f and %f is %f\n",a,b,c);

break;

case 'D':

case 'd': if(b==0)

printf("Division by zero\n");

else

{

c=a/b;

printf("The result of %f and %f is %f",a,b,c);

break;

}

default: printf(" \nUnknown operator\n");

break;

}

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc calc.c

meera@meera-laptop:~$ ./a.out

MENU

A:Addition

S:Subtration

M:Multiplication

D:Division

Enter your choice

A

Enter two numbers

2 3

The sum of 2.000000 and 3.000000 is 5.000000

meera@meera-laptop:~$ clear

meera@meera-laptop:~$ ./a.out

MENU

A:Addition

S:Subtration

M:Multiplication

D:Division

Enter your choice

A

Enter two numbers

2 3

The sum of 2.000000 and 3.000000 is 5.000000

meera@meera-laptop:~$ ./a.out

MENU

A:Addition

S:Subtration

M:Multiplication

D:Division

Enter your choice

S

Enter two numbers

2 3

The difference of 2.000000 and 3.000000 is -1.000000

meera@meera-laptop:~$ ./a.out

MENU

A:Addition

S:Subtration

M:Multiplication

D:Division

Enter your choice

m

Enter two numbers

4 5

The product of 4.000000 and 5.000000 is 20.000000

meera@meera-laptop:~$ ./a.out

MENU

A:Addition

S:Subtration

M:Multiplication

D:Division

Enter your choice

d

Enter two numbers

4 5

The result of 4.000000 and 5.000000 is 0.800000

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 8**

**SINE AND COSINE SERIES.**

**AIM** : Program to find the sum of the sine and cosine series:

Sine : X – X3 + X5 - X7 + ……

3! 5! 7!

Cosine : 1 - X2 + X4 - X6 + ……

2! 4! 6!

**INPUT SPECIFICATION :** x – value of the angle in degrees

**OUTPUT SPECIFICATION :** sum - sum of sine and cosine series

**ALGORITHM**

1. Start
2. Read the limit of the series
3. Read the value of x in degrees
4. x ← (x \* 3.14) / 180
5. sin\_term ← x
6. cos\_term ← x
7. Repet for I = 1,2,3,……………………..n

Begin

sum ← sum + term

term ← term \* (-x2 / 2i(2i+1))

cos\_sum ← cos \_sum + cos \_term

cos \_term ← cos \_term \* (-x2 / 2i(2i-1))

End

1. Stop

**PROGRAM DESCRIPTION**

Get the value of x in degrees .Convert it into radians

Using the series find the sum and compare it using the

sine and cosine function from the C library.

**PROGRAM**

#include<stdio.h>

#include<math.h>

int main()

{

int n,i,x,c=3,f=1,l,sign;

float sum;

printf(" enter the value of x and n\n");

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

sum=x;

for(i=3;i<=n;i+=2)

{

f=1;

if(c%2!=0)

{

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

f=f\*l;

sum=sum-pow(x,i)/f;

}

else

{

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

f=f\*l;

sum=sum+pow(x,i)/f;

}

c++;

}

printf(" Sum of sine series=%f", sum);

sign=1;

sum=1;

for(i=2;i<=n;i+=2)

{

f=1;

sign=sign\*-1;

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

f=f\*l;

sum=sum+sign\*pow(x,i)/f;

}

printf(" Sum of the cosine series= %f", sum);

return 0;

}

**OUTPUT**

student@user-desktop:~$ ./a.out

enter the value of x and n

2

5

Sum of sine series=0.933333

Sum of the cosine series= -0.333333

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 9**

**SIMPSON’S AND TRAPEZOIDAL METHOD**

**AIM** : Program to find the integral f(x) using Simpson’s and trapezoidal method.

**INPUT SPECIFICATION :** a and b- Values of a and b

**OUTPUT SPECIFICATION :** ans- Trapezoidal and simpson’s integral result.

**ALGORITHM**

1. Start
2. Read the two values- a and b.
3. h← (b-a)/n
4. y[0] ←F(a)
5. ans←y[0]
6. x[0] ←a
7. Begin
8. Repeat for I=1,2,3,4………………20

x[i] ←x[i-1]+h

y[i] ←F(x[i])

ans←ans+2\*y[i]

1. End.
2. y[n] ←F(b)
3. ans←ans+y[n]
4. ans←ans\*h/2
5. Print the ans- Trapezoidal
6. h← (b-a)/n
7. y[0] ←F(a)
8. ans←y[0]
9. x[0] ←a
10. Begin
11. Repeat for I=1,3,5………………19

x[i] ←x[i-1]+h

y[i] ←F(x[i])

ans←ans+4\*y[i]

1. End.
2. Repeat for I=2,4,6………………20

x[i] ←x[i-1]+h

y[i] ←F(x[i])

ans←ans+2\*y[i]

1. End.
2. y[n] ←F(b)
3. ans←ans+y[n]
4. ans←ans\*h/3
5. Print the ans- Simpson’s
6. Stop

**PROGRAM DESCRIPTION**

Finding the sum of integral f(x) with the help of simpson’s and trapezoidal rule.

**PROGRAM**

#include<stdio.h>

# define F(x) (1/(1+x))

#define n 10

int main()

{

float a,b,h,x[20],y[20],ans;

int i;

printf("enter the values of a and b\n");

scanf("%f%f",&a,&b);

/\* TRAPEZOIDAL METHOD\*/

h=(b-a)/n;

y[0]=F(a);

ans=y[0];

x[0]=a;

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

{

x[i] ←x[i-1]+h;

y[i] ←F(x[i]);

ans←ans+2\*y[i];

}

y[n] ←F(b);

ans←ans+y[n];

ans←ans\*h/2;

printf(" Trapezoidal answer=%f\n",ans);

/\* SIMPSON'S METHOD\*/

h=(b-a)/n;

y[0]=F(a);

ans=y[0];

x[0]=a;

for(i=1;i<n;i+=2)

{

x[i]=x[i-1]+h;

y[i]=F(x[i]);

ans=ans+4\*y[i];

}

for(i=2;i<n;i+=2)

{

x[i]=x[i-1]+h;

y[i]=F(x[i]);

ans=ans+2\*y[i];

}

y[n]=F(b);

ans=ans+y[n];

ans=ans\*h/3;

printf(" simpson's answer=%f\n",ans);

}

**OUTPUT**

student@user-desktop:~$ cc simpson\_trapezoidal.c

student@user-desktop:~$ ./a.out

enter the values of a and b

2

3

Trapezoidal answer=0.288690

simpson's answer=0.287698

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 10**

**selection sorting**

**AIM** : Program using function to sort a given unsorted array using selection sort method.

**INPUT SPECIFICATION :** n - The number of elements

a - The array to store the numbers

**OUTPUT SPECIFICATION :** Print the sorted list.

**ALGORITHM**

1. Start
2. Read the value of number of elements of the array, N
3. Read the array elements, A
4. Repeat for I = 1, 2, …………N-1

Begin

MIN← I

Repeat for J = 2, …………N

Begin

If a[J]<A[ MIN]

MIN = j

End

TEMP=A[I]

A[I]=A[M]

A[M]=TEMP

End

1. Print the sorted array
2. Stop

**program DESCRIPTION**

Program to find the list of unsorted array to a sorted one using selection sorting method.

**program**

#include<stdio.h>

int main()

{

int a[20],i,n,m,temp;

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

scanf("%d",&n);

printf("Enter the numbers\n");

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

{

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

}

printf(" The unsorted array is\n");

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

{

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

}

printf("\n");

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

{

m=min(a,n,i);

temp=a[i];

a[i]=a[m];

a[m]=temp;

}

printf("Sorted Array\n");

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

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

printf("\n");

return 0;

}

int min(int a[20],int n,int i)

{

int temp,min,j;

min=i;

for(j=i+1;j<n;j++)

if(a[j]<a[min])

min=j;

return min;

}

**OUTPUT**

student@user-desktop:~$ ./a.out

Enter the number of elements

6

Enter the numbers

43

21

34

67

42

1

The unsorted array is

43 21 34 67 42 1

Sorted Array

1 21 34 42 43 67

---------------------------------------------------------------------------------------------------------------------\*/

**cycle-111**

**EXPERIMENT NO : 11**

**NUMBER OF SENTENCES,WORDS ,LETTERS IN A LINE OF TEXT**

**AIM** : program to find the number of sentences, words, letter in the text

**INPUT SPECIFICATION :** text – the line of text

**OUTPUT SPECIFICATION :** sen\_count - Prints the number of sentences in the text

word\_count - Prints the number of words in the text

letter\_count - Prints the number of letters in the text

**ALGORITHM**

1. Start
2. Set sen\_count=0,word\_count=0,letter\_count=0
3. Get the line of text and store it in ‘text’
4. len = strlen(text)
5. i = 0
6. Is i < n

Then goto step 7

Else go to step 10

1. If text[i] is  **.** or **?**

Then sen\_count = sen\_count + 1

1. If text[i] is **space** or **.**  or  **\t** or **! ,**

Thenword\_count = word\_count + 1

1. If ( text[i] >= 65 **and** text[i] <= 90) **or** ( text[i] >= 97 **and** text[i] <= 122))

Then letter\_count = letter\_count + 1

1. Print sen\_count , word\_count and letter\_count
2. Stop

**PROGRAM DESCRIPTION**

sentence count - checks for ? and .

word count - checks for ? . , ! \t and space

letter count - Checks the ASCII value

**PROGRAM**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char text[100];

int i,len;

int sen\_count=0,word\_count=0,letter\_count=0;

clrscr();

printf("\n\t\tCounts the no: of sentences, words and letters\n");

printf("\t\t----------------------------------------------\n");

printf("\n\nEnter a line of text : ");

scanf("%[^\n]s",&text);

len = strlen(text);

//Computes the number of sentences,words and letters.

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

{

if(text[i] == '.' || text[i] == '?')

sen\_count = sen\_count + 1;

if(text[i] == ' ' || text[i] == '.' || text[i] == '\t' ||

text[i] == '!' || text[i] == ','|| text[i] == '?')

word\_count = word\_count + 1;

if((text[i] >= 65 && text[i] <= 90) ||

(text[i] >= 97 && text[i] <= 122))

letter\_count = letter\_count + 1;

}

printf("\nThe number of : \n\n");

printf("\nSentences : %d\n",sen\_count);

printf("\nWords : %d\n",word\_count);

printf("\nLetters : %d\n",letter\_count);

getch();

}//End of program

**OUTPUT**

Counts the no: of sentences, words and letters

-------------------------------------------------------------

Enter a line of text : Hi Ram,How are you?I am fine,Thank you.

The number of :

Sentences : 2

Words : 10

Letters : 29

**EXPERIMENT NO : 12**

**SORTING of names**

**AIM** : Program to read ‘n’ names and to sort them in alphabetical order.

**INPUT SPECIFICATION :** limit - The number of elements

name - The array to store the names

**OUTPUT SPECIFICATION :** print the number in the sorted order

**ALGORITHM**

1. Start
2. Read the value of number of names, limit
3. Read the array names, A
4. Repeat for I= 1, 2, …………N-1

Begin

Repeat for J = 1, 2,…………….N-1

Begin

IF name[I] > name[I+1] THEN

Begin

TEMP = name[I]

name[I] = name[I+1]

name[I+1] = TEMP

End

End

End

1. Print “ The Sorted names are “
2. Repeat for I = 1, 2, …………………….N

Begin

Print name[I]

End

1. Stop

**PROGRAM DESCRIPTION**

Program to sort n names in ascending order using bubble sort

**PROGRAM**

#include<stdio.h>

#include<string.h>

int main()

{

char name[10][10],temp[20];

int i,j,limit;

printf("Enter the upper limit not more than 10\n");

scanf("%d",&limit);

printf("Enter the names\n");

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

{

scanf("%s",name[i]);

}

printf(" The unsorted array of names are\n");

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

{

puts(name[i]);

printf("\n");

}

printf("The sorted array of names are\n");

for(j=0;j<limit-1;j++)

{

for(i=0;i<limit-1;i++)

if(strcmp(name[i],name[i+1]) >0)

{

strcpy(temp,name[i]);

strcpy(name[i],name[i+1]);

strcpy(name[i+1],temp);

}

}

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

printf("%s\n", name[i]);

printf("\n\n");

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc name\_sort.c

meera@meera-laptop:~$ ./a.out

Enter the upper limit not more than 10

6

Enter the names

meera

meena

deepa

deepu

seena

teena

The unsorted array of names are

meera

meena

deepa

deepu

seena

teena

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 13**

**MATRIX ADDITION aND differenCe**

**AIM** : Program to add and difference of two matrix.

**INPUT SPECIFICATION :** (r1,r2) The order of the matrix

A , B ,the input matrices whose sum and difference has to be found

**OUTPUT SPECIFICATION :** C , the output matrix which stores the sum and difference Aand B

**ALGORITHM**

1. Start
2. Read the order fo the matrics (r1,c1)
3. READ\_MATRIX A

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Read a[i][j]

End

End

1. READ\_MATRIX B

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Read b[i][j]

End

End

1. WRITE\_MATRIX A

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Write a[i][j]

End

End

1. WRITE\_MATRIX B

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Write b[i][j]

End

End

1. ADD\_MATRIX

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

C[i][j] ← A[i][j] + B[i][j]

End

End

1. SUBTRACT MATRIX

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

C[i][j] ← A[i][j] - B[i][j]

End

End

1. Print matrix C
2. Stop

**PROGRAM DESCRIPTION**

Addition and subtraction of two matrices a and b**.**

**PROGRAM**

#include<stdio.h>

int main()

{

int i,j,a[10][10],b[10][10],c[10][10],r1,c1;

printf("Enter the order of Matrix A and B up to 10\*10:\n");

scanf("%d%d",&r1,&c1);

printf(" Enter the elements in Matrix A\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

}

printf(" Enter the elements in Matrix B\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

}

printf(" The elements in Matrix A\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

printf("\n");

}

printf(" The elements in Matrix B\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

printf("%d\t",b[i][j]);

printf("\n");

}

printf(" Matrix Addition\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

c[i][j]=a[i][j]+b[i][j];

}

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

printf("%d\t",c[i][j]);

printf("\n");

}

printf(" Matrix Subtraction\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

c[i][j]=a[i][j]-b[i][j];

}

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

printf("%d\t",c[i][j]);

printf("\n");

}

return 0;

}

**OUTPUT**

student@user-desktop:~$ cc sum\_diff\_marti.c

student@user-desktop:~$ ./a.out

Enter the order of Matrix A and B up to 10\*10:

3

3

Enter the elements in Matrix A

3

3

4

5

6

7

8

9

1

Enter the elements in Matrix B

3

4

5

6

7

8

9

3

4

The elements in Matrix A

3 3 4

5 6 7

8 9 1

The elements in Matrix B

3 4 5

6 7 8

9 3 4

Matrix Addition

6 7 9

11 13 15

17 12 5

Matrix Subtraction

0 -1 -1

-1 -1 -1

-1 6 -3

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 14**

**MATRIX MULTIPLICATION**

**AIM** : Program to multiply two matrix.

**INPUT SPECIFICATION :** (r1,c1) The order of the matrix A

(r2,c2) The order of matrix B

A , B ,the input matrices whose product has to be found

**OUTPUT SPECIFICATION :** C , the output matrix which stores the product of Aand B

**ALGORITHM**

1. Start
2. Read the order of the matrics (r1,c1) A
3. Read the order of the matrices (r2,c2) B
4. READ\_MATRIX A

Begin

Repeat for I = 1,2,………………….r1

Begin

Repeat for J = 1,2,………………….c1

Read a[i][j]

End

End

1. READ\_MATRIX B

Begin

Repeat for I = 1,2,………………….r2

Begin

Repeat for J = 1,2,………………….c2

Read b[i][j]

End

End

1. WRITE\_MATRIX A

Begin

Repeat for I = 1,2,………………….r1

Begin

Repeat for J = 1,2,………………….c2

Write a[i][j]

End

End

1. WRITE\_MATRIX B

Begin

Repeat for I = 1,2,………………….r2

Begin

Repeat for J = 1,2,………………….c2

Write b[i][j]

End

End

8. Repeat step 9 until i<r1,j<c2,k<c1

9.d[i][j]=0;

10. c[i][j]=c[i][j]+a[i][k]\*b[k][j]

11.Set a loop to print matrix values.

12.Print matrix value c[i][j]

13. Stop.

**PROGRAM DESCRIPTION**

Product of two matrices a and b**.**

**PROGRAM**

#include<stdio.h>

int main()

{

int i,j,a[10][10],b[10][10],c[10][10],r1,c1,r2,c2,sum,k;

printf("Enter the order of Matrix A up to 10\*10:\n");

scanf("%d%d",&r1,&c1);

printf("Enter the order of Matrix B up to 10\*10:\n");

scanf("%d%d",&r2,&c2);

printf(" Enter the elements in Matrix A\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

}

printf(" Enter the elements in Matrix B\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

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

}

printf(" The elements in Matrix A\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

printf("\n");

}

printf(" The elements in Matrix B\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

printf("%d\t",b[i][j]);

printf("\n");

}

if(c1!=r1)

{

printf("Matrix cannot be muliplicable\n");

//exit(1);

}

printf(" Matrix Multiplication\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

sum=0;

for(k=0;k<c1;k++)

sum+=a[i][k]\*b[k][j];

c[i][j]=sum;

}

}

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

printf("%d\t",c[i][j]);

printf("\n");

}

return 0;

}

**OUTPUT**

meera@meera-laptop:~$ cc product\_martix.c

meera@meera-laptop:~$ ./a.out

Enter the order of Matrix A up to 10\*10:

3

3

Enter the order of Matrix B up to 10\*10:

3

3

Enter the elements in Matrix A

1

2

3

4

5

6

7

8

9

Enter the elements in Matrix B

9

8

7

6

5

4

3

2

1

The elements in Matrix A

1 2 3

4 5 6

7 8 9

The elements in Matrix B

9 8 7

6 5 4

3 2 1

Matrix Multiplication

30 24 18

84 69 54

138 114 90

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 15**

**Searching using binary search**

**AIM** : Program to search an integer in the unsorted array using binary search.

**INPUT SPECIFICATION :** n - The number of elements

a - The array of elements

key- Element to be searched

**OUTPUT SPECIFICATION :** print the successful if the element is there in the list

Print unsuccessful if element is not found

**ALGORITHM**

1. Start
2. Read the value of number of elements, n
3. Read the array names, a
4. Repeat for I= 1, 2, …………N-1

Begin

Repeat for J = 1, 2,…………….N-1

Begin

IF a[I] > a[I+1] THEN

Begin

TEMP = a[I]

a[I] = a[I+1]

a[I+1] = TEMP

End

End

End

1. high=n-1
2. low=0
3. Repeat for low<=high

Begin

mid=(low+high)/2;

if key=a[mid]

print success

else key>a[mid]

low=mid+1

else

high=mid-1

End

1. Print Unsuccessful if the key not found in a
2. Stop

**PROGRAM DESCRIPTION**

Program to search an element using binary search method

**PROGRAM**

#include<stdio.h>

int main()

{

int i,n,a[20],low,high,mid,key,j,temp;

printf("Enter the value for n\n");

scanf("%d",&n);

printf(" Enter the %d elements\n",n);

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

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

printf(" enter the element to be searched\n");

scanf("%d", &key);

printf(" The Elements are \n");

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

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

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

{

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

{

if(a[i]>a[i+1])

{

temp=a[i];

a[i]=a[i+1];

a[i+1]=temp;

}

}

}

printf("\n\n The Sorted lists are\n");

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

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

high=n-1;

low=0;

while(low<=high)

{

mid=(low+high)/2;

if(key==a[mid])

break;

else if(key>a[mid])

low=mid+1;

else

high=mid-1;

}

if(a[mid]==key)

{

printf(" \n\nSearch Successful!!\n");

printf("%d is found in position number %d\n" ,key,mid+1);

}

else

printf(" Search is unsuccessful, %d not found\n", key);

return 0;

}

**OUTPUT**

student@user-desktop:~$ cc binarysearch.c

student@user-desktop:~$ ./a.out

Enter the value for n

7

Enter the 7 elements

3

41

8

9

4

3

2

enter the element to be searched

41

The Elements are

3 41 8 9 4 3 2

The Sorted lists are

2 3 3 4 8 9 41

Search Successful!!

41 is found in position number 7

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 16**

**sum of all elements in array using pointers**

**AIM** : Program to find the sum of all elements in an array using poinetrs.

**INPUT SPECIFICATION :** N - the number of elements in the array

A – the array elements

**OUTPUT SPECIFICATION :** sum- sum of all elements

**ALGORITHM**

1. Start
2. Read the number of elements of the array N
3. Read the array elements A
4. \*p ← A
5. Repeat for I = 1,2,…………………..upto N

Begin

sum=sum+\*p

End

1. Print sum
2. Stop

**PROGRAM DESCRIPTION**

Sum of all the elements in an array using pointers.

**PROGRAM**

#include<stdio.h>

int main()

{

int i,n,sum=0,a[10];

int \*p;

printf(" Enter the value of n\n");

scanf("%d",&n);

printf(" Enter %d elements to array a\n", n);

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

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

printf(" The address of a");

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

printf("= %u\n", &a[i]);

printf(" The elements are\n");

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

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

for(p=a;p<a+n;p++)

sum+=\*p;

printf(" \n Sum = %d\n", sum);

return 0;

}

**OUTPUT**

student@user-desktop:~$ cc sum\_pointers.c

student@user-desktop:~$ ./a.out

Enter the value of n

4

Enter 4 elements to array a

1

2

3

4

The address of a= 3214194172

= 3214194176

= 3214194180

= 3214194184

The elements are

1

2

3

4

Sum = 10

---------------------------------------------------------------------------------------------------------------------\*/

**CYCLE - IV**

**EXPERIMENT NO : 16**

**swap two variables using pointers**

**AIM** : Program to swap the contents of two variables using pointers and function.

**INPUT SPECIFICATION :** a and b- two elements to be swapped

**OUTPUT SPECIFICATION :** print the contents of a and b after swapping

**ALGORITHM**

1. Start
2. Read a,b – the numbers for swapping
3. Print the values of a and b before swapping
4. Call the function swap(&a,&b)
5. Print the values of a and b after swapping

6. Stop

// swap function

1. Declaring a variable temp,temporary variable
2. temp = \*x
3. \*x = \*y
4. \*y = temp
5. return

**PROGRAM DESCRIPTION**

Swap the contents of two variables using pointers.

**PROGRAM**

#include<stdio.h>

int main()

{

void swap(int \*x,int \*y);

int a,b;

printf("Enter the values of a and b\n");

scanf("%d%d",&a,&b);

printf("\n\nThe values of a a and b before swapping a=%d and b=%d\n",a,b);

swap(&a,&b);

printf("\nThe values of a a and b after swapping a=%d and b=%d\n",a,b);

}

void swap(int \*x,int \*y)

{

int temp;

temp=\*x;

\*x=\*y;

\*y=temp;

return;

}

**OUTPUT**

student@user-desktop:~$ cc swap\_fn\_pointer.c

student@user-desktop:~$ ./a.out

Enter the values of a and b

23

56

The values of a a and b before swapping a=23 and b=56

The values of a a and b after swapping a=56 and b=23

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 17**

**create linked list, insert the element, add a new element and deletion**

**AIM** : Program to create a linked list, insert the element, Add a new element to a position and delete the element from the linked list. Also display and count the elements in the linked list.

**INPUT SPECIFICATION :** info- element to be inserted

Choice**-** what operation to be performed- insertion, display, deletion or counting

**OUTPUT SPECIFICATION :** print the contents in the linked list and also the count the number of nodes

**ALGORITHM**

1. Start
2. Start with the first node.
3. Repeat

Print the current item

Advance to the next node.

1. If the list is empty

Insert the new node as the head node.

1. Else

Insert the new node as the last node

1. Else

Insert the new node in the body of the list.

1. For deleteting a node form the list check if the list is empty then,

Node cannot be deleted

Else

If the node to be deleted is the first node then make the head to point to the second node

Else

Delete the node from the body of the list.

1. Display all the values of the node and display the counted number of nodes nodes.

6. Stop

**PROGRAM**

#include<stdio.h>

#include<malloc.h>

#include<stdlib.h>

#define NULL 0

struct list\_element

{

int item;

struct list\_element \*next;

};

typedef struct list\_element node;

int main()

{

node \*first;

int key,c;

int choice;

int menu();

node \*create();

node \*insert();

node \*delete();

void display();

int count();

do

{

choice=menu();

switch(choice)

{

case 1:

printf("Press -999 to STOP\n");

first=create(first);

printf("\n\n Created list\n");

display(first);

continue;

case 2:

first=insert(first);

printf("\nLIST\n");

display(first);

continue;

case 3:

first=delete(first);

continue;

case 4:

c=count(first);

printf("Number of elements in the list=%d",c);

continue;

default:

printf("\nEnd of Execution\n");

}

}while(choice!=5);

}

int menu()

{

int choice;

do

{

printf("\n\n Main Menu\n");

printf("1- Create the linked list\n");

printf("2- Insert an item\n");

printf("3- Delete an item\n");

printf("4-Counting number\n");

printf("5- End\n");

printf(" Enter your Choice (1,2,3,4,5) : \n");

scanf("%d",&choice);

if(choice<1 || choice>5)

printf("\n Invalid Choice-try again\n");

}while(choice<1 || choice>5);

printf("\n");

return(choice);

}

/\*creating a linked list\*/

node \*create(first)

node \*first;

/\*first point to the current node\*/

{

int info;

node \*temp,\*prev;

prev=first=NULL;

printf("\n Data Item\n");

scanf("%d",&info);

while(info!=-999)

{ /\*Allocate memory space for the next node\*/

temp=(node \*)malloc(sizeof(node));

temp->item=info;

temp->next=NULL;

if(first==NULL)

first=temp;

else

prev->next=temp;

prev=temp;

printf("Data Item:\n");

scanf("%d",&info);

//printf("%d",info);

}

return(first);

}

/\*Display the linked list recursively\*/

void display(first)

node \*first;

{

int c=0;

if(first!=NULL)

{

printf("->%d",first->item);

display(first->next);

c++;

}

return ;

}

/\*Add one element to linked list and return pointer to beginning of modified list\*/

node \*insert(first)

node \*first;

/\* first points to first node\*/

{

node \*newnode;

node \*temp;

int newitem;

int position;

int i;

printf(" New data item\n");

scanf("%d",&newitem);

do

{

printf("\n Position of insertion\n");

scanf("%d",&position);

}while(position<=0);

if((position==1) || (first==NULL))

{

newnode=(node \*)malloc(sizeof(node));

newnode->item=newitem;

newnode->next=first;

first=newnode;

}

else

{ i=1;

temp=first;

while((i<position-1)&&(temp->next!=NULL))

{

temp=temp->next;

i++;

}

newnode=(node \*)malloc(sizeof(node));

newnode->item=newitem;

newnode->next=temp->next;

temp->next=newnode;

}

return(first);

}

/\*delete one omponent fom linked list\*/

node \*delete(first)

node \*first;

{

node \*temp;

node \*prev;

int target;

printf("Data item to be deleted\n");

scanf("%d",&target);

if(first==NULL)

printf(" LIST IS EMPTY- CANNOT DELETE\n");

else

{ prev=NULL;

temp=first;

while((temp!=NULL)&&(temp->item!=target))

{

prev=temp;

temp=temp->next;

}

if(temp==NULL)

printf("Element not found\n");

else

{

if(prev==NULL)

first=first->next;

else

prev->next=temp->next;

printf("\nLIST\n");

display(first); }

}

return(first);

}

int count(first)

node \*first;

{

int c=0;

while(first!=NULL)

{

first=first->next;

c++;

}

return c;

}

**OUTPUT**

student@user-desktop:~$ ./a.out

Main Menu

1- Create the linked list

2- Insert an item

3- Delete an item

4-Counting number

5- End

Enter your Choice (1,2,3,4,5) :

1

Press -999 to STOP

Data Item

2

Data Item:

3

Data Item:

8

Data Item:

9

Data Item:

4

Data Item:

-999

Created list

->2->3->8->9->4

Main Menu

1- Create the linked list

2- Insert an item

3- Delete an item

4-Counting number

5- End

Enter your Choice (1,2,3,4,5) :

2

New data item

67

Position of insertion

6

LIST

->2->3->8->9->4->67

Main Menu

1- Create the linked list

2- Insert an item

3- Delete an item

4-Counting number

5- End

Enter your Choice (1,2,3,4,5) :

4

Number of elements in the list=6

Main Menu

1- Create the linked list

2- Insert an item

3- Delete an item

4-Counting number

5- End

Enter your Choice (1,2,3,4,5) :

3

Data item to be deleted

2

LIST

->3->8->9->4->67

Main Menu

1- Create the linked list

2- Insert an item

3- Delete an item

4-Counting number

5- End

Enter your Choice (1,2,3,4,5) :

5

End of Execution

-------------------------------------------------------------------------------------------------------------------------

**EXPERIMENT NO : 18**

**NUMBER OF chars, spaces, tabs and new lines in a file**

**AIM** : program to find the number of chars, spaces, tabs and new lines in a file

**INPUT SPECIFICATION :** fp – file having the input

**OUTPUT SPECIFICATION :** noc - Prints the number of characters in the file

nob - Prints the number blanks in the file

not - Prints the number of tabs in the file

nol- print number of lines in the file

**ALGORITHM**

1. Start
2. Set noc=0 nob=0,nol=0,not=0
3. Get the line of text and store it in ‘text’
4. len = strlen(text)
5. i = 0
6. Is i < n

Then goto step 7

Else go to step 10

1. If text[i] is  **‘ ‘**

Then nob = nob+ 1

1. If text[i] is **‘\n’ ,**

Then **nol** = nolt+ 1

1. If text[i] is ‘\t’

Then not= not+ 1

1. Print noc, not, nol, nob
2. Stop

**PROGRAM**

#include<stdio.h>

int main()

{

FILE \*fp;

char ch;

int nol=0, not=0, nob=0, noc=0;

fp= fopen(“PR1,c”, “r”);

while(1)

{

ch= fgetc(fp);

If(ch== EOF)

break;

noc++;

if(ch==’ ‘)

nob++;

if(ch==’\n’)

nol++;

if(ch==’\t’)

not++;

}

Fclose(fp);

printf(“ Number of characters= %d\n”, noc);

printf(“ Number of blanks= %d\n”, nob);

printf(“ Number of tabs= %d\n”, not);

printf(“ Number of lines= %d\n”, nol);

return 0;

}

**OUTPUT**

Number of characters=125

Number of blanks= 25

Number of tabs=13

Number of lines=22

---------------------------------------------------------------------------------------------------------------------\*/

**EXPERIMENT NO : 19**

**AIM** : To find the details of the students using structure

**ALGORITHM**

1. Start
2. Create a structure student with the details rollno,studname,sub1,sub2,sub3,total marks and percentage
3. Read the student details
4. List the studname who have scored more than 60% marks in 3 subjects.
5. Print the details
6. Stop

**PROGRAM DESCRIPTION**

Read the details of student,calculate percentage and list the studname who have scored more than 60% marks in 3 subjects.

**PROGRAM**

#include<stdio.h>

struct student

{

char name[30];

int sub1;

int sub2;

int sub3;

int total;

float percent;

};

int main()

{

int i,n;

struct student s[20];

printf(“Enter the number of students\n”);

scanf(“%d”,&n);

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

{

printf(“Enter the student name\n”);

gets(s[i].name);

printf(“Enter the marks of sub1 and sub2 and sub 3\n”);

scanf(“%d”,&s[i].sub1);

scanf(“%d”,&s[i].sub1);

scanf(“%d”,&s[i].sub1);

}

Printf(“List of Students name who have scored more than 60% of marks\n\n”);

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

{

s[i].total=s[i].sub1+s[i].sub2+s[i].sub3;

s[i].percent=s[i].total/3.0;

if(s[i].percent>=60)

printf(“%s\t%f\t”,s[i].name,s[i].percent);

}

}

**OUTPUT**

Enter the number of students

2

Enter the student name

Ann

Enter the marks of sub1 and sub2 and sub 3

80

80

80

Enter the student name

Binu

Enter the marks of sub1 and sub2 and sub 3

20

34

21

List of Students name who have scored more than 60% of marks

Ann 124

**EXPERIMENT NO : 2**

**MEAN VARIANCE STANDARD DEVIATION**

**AIM** : Program to calculate the mean, variance and deviation of a given set of numbers.

**INPUT SPECIFICATION :** n - The number of elements

a - The array to store the numbers

**OUTPUT SPECIFICATION :** Print mean , variance and standard deviation

**ALGORITHM**

1. Start
2. Read the value of number of elements, N
3. Read the array elements, A
4. Set MEAN ← 0
5. Set VARIANCE ← 0
6. Set STDDEV ← 0
7. Repeat for I = 1, 2, …………N

Begin

SUM = SUM + A[I]

End

1. MEAN = SUM/N
2. SUM ← 0
3. Repeat for I = 1, 2, …………..N

Begin

SUM = SUM + (A[I] – MEAN)2

End

1. VARIANCE = SUM/N
2. STDDEV = √ (VARIANCE)
3. Print “ Mean =”, MEAN
4. Print “ Std Deviation =”, STDDEV
5. Print “ Variance =”, VARIANCE
6. Stop

**PROGRAM DESCRIPTION**

Program to calculate the mean,variance and stddev of

a given set of numbers.

**PROGRAM**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int n,i;

float sum = 0, a[25];

float mean,variance,stddev;

clrscr();

printf("\n\t\tMEAN VARIANCE AND STANDARD DEVIATION\n");

printf("\t\t------------------------------------\n");

printf("\nEnter the value of number of elements : ");

scanf("%d",&n);

printf("\nEnter the array elements : ");

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

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

//Calculate mean

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

sum = sum + a[i];

mean = sum/n;

//Calculate variance

sum = 0;

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

sum += (a[i]-mean) \* (a[i]-mean);

variance = sum/n;

//Calculate standard deviation

stddev=sqrt(variance);

printf("\nMean = %f",mean);

printf("\nVariance = %f",variance);

printf("\nStandard deviation = %f",stddev);

getch();

}

**OUTPUT**

MEAN VARIANCE AND STANDARD DEVIATION

------------------------------------------------------------------

Enter the value of number of elements : 5

Enter the array elements : 10 20 30 40 50

Mean = 30.000000

Variance = 200.000000

Standard deviation = 14.142136

**EXPERIMENT NO : 3**

**MAXIMUM AND MINIMUM USING LINEAR SEARCH**

**AIM** : Program to find the smallest and largest numbers in an array of integers using

linear search.

**INPUT SPECIFICATION :** n - The number of elements

a - The array to store the numbers

**OUTPUT SPECIFICATION :** Print the smallest and the largest number

**ALGORITHM**

1. Start
2. Read the value of number of elements of the array, N
3. Read the array elements, A
4. Set MIN ← A[I]
5. Set MAX ← A[I]
6. Set POS\_MAX ← 1
7. Set POS\_MIN ← 1
8. Repeat for I = 1, 2, …………N

Begin

IF MIN > A[I] THEN

Begin

MIN = A[I]

POS\_MIN = I

End

IF MAX < A[I] THEN

Begin

MAX = A[I]

POS\_MAX = I

End

End

1. Print “Minimum = “, MIN
2. Print “Position of minimum = “, POS-MIN
3. Print “Maximum = “, MAX
4. Print “Minimum = “, MIN
5. Stop

**program DESCRIPTION**

Program to find the maximum and minimum among given n integer numbers using linear search method.

**program**

#include<stdio.h>

#include<conio.h>

void main()

{

int n,i,min=0,max=0,pos\_min=0,pos\_max=0,a[25];

clrscr();

printf("\n\t\tMaximum and Minimum\n");

printf("\t\t-------------------\n");

printf("\nEnter the value of the number of elements in an array : ");

scanf("%d",&n);

printf("\nEnter the array elements : ");

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

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

/\*Initially the first element is assumed to be both minimum(min) as

well as maximum(max) \*/

min=a[1];

max=a[1];

pos\_min=1;

pos\_max=1;

// To find the minimum as well as maximum

for(i=2;i<=n;i++)

{

if(min > a[i])

{

min=a[i];

pos\_min=i;

}

if(max < a[i])

{

max=a[i];

pos\_max=i;

}

}

printf("\nMinimum = %d \tPosition = %d\n",min,pos\_min);

printf("\nMaximum = %d \tPosition = %d",max,pos\_max);

getch();

}

**OUTPUT**

Maximum and Minimum

--------------------------------

Enter the value of the number of elements in an array : 7

Enter the array elements : 13 74 95 2 17 45 32

Minimum = 2 Position = 4

Maximum = 95 Position = 3

**EXPERIMENT NO : 4**

**SORTING USING BUBBLE SORT**

**AIM** : Program to sort n numbers in ascending order using bubble sort.

**INPUT SPECIFICATION :** n - The number of elements

a - The array to store the numbers

**OUTPUT SPECIFICATION :** print the number in the sorted order

**ALGORITHM**

1. Start
2. Read the value of number of elements, N
3. Read the array elements, A
4. Repeat for J = 1, 2, …………N-1

Begin

Repeat for I = 1, 2,…………….N-1

Begin

IF A[I] > A[I+1] THEN

Begin

TEMP = A[I]

A[I] = A[I+1]

A[I+1] = TEMP

End

End

End

1. Print “ The Sorted elements are “
2. Repeat for I = 1, 2, …………………….N

Begin

Print A[I]

End

1. Stop

**PROGRAM DESCRIPTION**

Program to sort n numbers in ascending order using bubble sort

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void main()

{

int n,i,j,a[25],temp=0;

clrscr();

printf("\n\t\tSorting using Bubble sort\n");

printf("\t\t-------------------------\n");

printf("\nEnter the value of the number of elements in the array : ");

scanf("%d",&n);

printf("\nEnter the array elements : ");

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

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

//Sorting using bubble sort

//This loop performs required number of passes

for(j=1;j<n;j++)

{

//This loop performs all comparisons in a pass

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

{

//Sorts if two consecutive numbers are not in the required order

if( a[i] > a[i+1] )

{

temp = a[i];

a[i] = a[i+1];

a[i+1] = temp;

}

}

}

// Sorted elements

printf("\nThe sorted elements are .......\n\n");

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

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

getch();

}

**OUTPUT**

Sorting using Bubble sort

----------------------------------

Enter the value of the number of elements in the array : 5

Enter the array elements : 5 4 3 2 1

The sorted elements are .......

1 2 3 4 5

**EXPERIMENT NO : 5**

**NUMBER OF SENTENCES,WORDS ,LETTERS IN A LINE OF TEXT**

**AIM** : program to find the number of sentences, words, letter in the text

**INPUT SPECIFICATION :** text – the line of text

**OUTPUT SPECIFICATION :** sen\_count - Prints the number of sentences in the text

word\_count - Prints the number of words in the text

letter\_count - Prints the number of letters in the text

**ALGORITHM**

1. Start
2. Set sen\_count=0,word\_count=0,letter\_count=0
3. Get the line of text and store it in ‘text’
4. len = strlen(text)
5. i = 0
6. Is i < n

Then goto step 7

Else go to step 10

1. If text[i] is  **.** or **?**

Then sen\_count = sen\_count + 1

1. If text[i] is **space** or **.**  or  **\t** or **! ,**

Thenword\_count = word\_count + 1

1. If ( text[i] >= 65 **and** text[i] <= 90) **or** ( text[i] >= 97 **and** text[i] <= 122))

Then letter\_count = letter\_count + 1

1. Print sen\_count , word\_count and letter\_count
2. Stop

**PROGRAM DESCRIPTION**

sentence count - checks for ? and .

word count - checks for ? . , ! \t and space

letter count - Checks the ASCII value

**PROGRAM**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char text[100];

int i,len;

int sen\_count=0,word\_count=0,letter\_count=0;

clrscr();

printf("\n\t\tCounts the no: of sentences, words and letters\n");

printf("\t\t----------------------------------------------\n");

printf("\n\nEnter a line of text : ");

scanf("%[^\n]s",&text);

len = strlen(text);

//Computes the number of sentences,words and letters.

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

{

if(text[i] == '.' || text[i] == '?')

sen\_count = sen\_count + 1;

if(text[i] == ' ' || text[i] == '.' || text[i] == '\t' ||

text[i] == '!' || text[i] == ','|| text[i] == '?')

word\_count = word\_count + 1;

if((text[i] >= 65 && text[i] <= 90) ||

(text[i] >= 97 && text[i] <= 122))

letter\_count = letter\_count + 1;

}

printf("\nThe number of : \n\n");

printf("\nSentences : %d\n",sen\_count);

printf("\nWords : %d\n",word\_count);

printf("\nLetters : %d\n",letter\_count);

getch();

}//End of program

**OUTPUT**

Counts the no: of sentences, words and letters

-------------------------------------------------------------

Enter a line of text : Hi Ram,How are you?I am fine,Thank you.

The number of :

Sentences : 2

Words : 10

Letters : 29

**EXPERIMENT NO : 6**

**MATRIX ADDITION aND mULTIPLICATION USING FUNCTION**

**AIM** : Program to add two matrices using function.

**INPUT SPECIFICATION :** (m,n) The order of the matrix

A , B ,the input matrices whose sum has to be found

**OUTPUT SPECIFICATION :** C , the output matrix which stores the sum of matrix Aand B

**MATRIX ADDITION**

**ALGORITHM**

1. Start
2. Read the order fo the matrics (m,n)
3. Call function READ\_MATRIX (A,m.n)
4. Call function READ\_MATRIX (B,m.n)
5. Call function ADD\_MATRIX (A,B,C,m.n)
6. Call function WRITE\_MATRIX (A,m.n)
7. Call function WRITE \_MATRIX (B,m.n)
8. Call function WRITE \_MATRIX (C,m.n)
9. Stop

READ\_MATRIX(mat[][10],m,n)

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Read mat[i][j]

End

End

WRITE\_MATRIX(mat[][10],m,n)

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

Write mat[i][j]

End

End

ADD\_MATRIX(A[][10], B[][10], C[][10],m,n)

Begin

Repeat for I = 1,2,………………….m

Begin

Repeat for J = 1,2,………………….n

C[i][j] ← A[i][j] + B[i][j]

End

End

**PROGRAM DESCRIPTION**

Functions read\_martix() and write\_matrix() are used to input and output the input matrices and sum of the matrices. The user defined function add\_matrix computes the sum of the two matrices.

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void main()

{

//Declaration

int a[10][10],b[10][10],c[10][10];

int m,n;

//Function prototypes

void read\_matrix(int mat[][10],int m , int n);

void write\_matrix(int mat[][10],int m , int n);

void add\_matrix(int a[][10],int b[][10],int c[][10],int m , int n);

clrscr();

printf("\n\t\tADDITION OF MATRICES\n");

printf("\t\t--------------------\n");

printf("\n\nEnter the order of the matrices A and B : ");

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

printf("\nEnter the elements of matrix A \n\n");

read\_matrix(a,m,n); //Calling the function 'read\_matrix'

printf("\nEnter the elements of matrix B \n\n");

read\_matrix(b,m,n);

add\_matrix(a,b,c,m,n);

//Prints matrix A , B and C using the function write\_matrix

printf("\n\nMatrix A................\n\n");

write\_matrix(a,m,n);

printf("\n\nMatrix B................\n\n");

write\_matrix(b,m,n);

printf("\n\nSum of A and B .........\n\n");

write\_matrix(c,m,n);

getch();

}//End of main function

//Function to read a matix

void read\_matrix(int mat[][10],int m ,int n)

{

int i,j;

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

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

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

return;

}

//Function to print a matix

void write\_matrix(int mat[][10],int m , int n)

{

int i,j;

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

{

printf("\n");

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

printf(" %d ",mat[i][j]);

}

return;

}

//Computes the sum of the matrices A and B

void add\_matrix(int a[][10],int b[][10],int c[][10],int m , int n)

{

int i,j;

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

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

c[i][j] = a[i][j] + b[i][j];

return;

}

**OUTPUT**

ADDITION OF MATRICES

-----------------------------------

Enter the order of the matrices A and B : 3 3

Enter the elements of matrix A

1 2 3 4 5 6 7 8 9

Enter the elements of matrix B

9 8 7 6 5 4 3 2 1

Matrix A................

1 2 3

4 5 6

7 8 9

Matrix B................

9 8 7

6 5 4

3 2 1

Sum of A and B .........

10 10 10

10 10 10

10 10 10

**MATRIX MULTIPLICATION**

**ALGORITHM:**

1.Start fuction

2.Read rows and columns limit for matrices m,n,p,q

3.Check p is equal to n else goto step 12

4.Set a loop to get A matrix values.

5.Read matrix value a[i][j]

6.Set a loop to get B matrix

7.Read matrix value b[i][j]

8.Repeat step 9 until i<m,j<n,k<p

9.d[i][j]=0;

10. d[i][j]=d[i][j]+a[i][k]\*b[k][j]

11.Set a loop to print matrix values.

12.Print matrix value d[i][j] goto step 13

13.Print the number of rows and columns should not be equal

14.Stop the fuction

**PROGRAM**

#include <stdio.h>

#include <conio.h>

void main ()

{

int a[10][10],b[10][10],d[10][10];

int i,j,p,q,m,n,k;

clrscr ();

printf ("Enter the size of the A matrix:");

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

printf ("Enter the size of the B matrix:");

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

if (p==n)

{

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

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

{

for (j=0;j<q;j++)

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

}

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

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

{

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

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

}

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

{

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

{

d[i][j]=0;

for(k=0;k<p;k++)

d[i][j]=d[i][j]+a[i][k]\*b[k][j];

}

}printf ("Multiplication of A andB matrix:\n");

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

{

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

printf ("%5d",d[i][j]);

printf ("\n");

}

}

else

printf ("The no. of rows and columns should not be equal");

getch();

}

**OUTPUT**

Enter the size of the A matrix:2 2

Enter the size of the B matrix:2 2

Enter the elements of A matrix.

1 2 3 4

Enter the elements of B matrix.

12 2 3 4

Multiplication of A andB matrix:

18 10

48 22

**EXPERIMENT NO : 10**

**nCr USING FUNCTION**

**AIM** : A program to find nCr using function

n!

nCr = -------------

r! (n-r)!

**INPUT SPECIFICATION :** Get the values of n and r

**OUTPUT SPECIFICATION :** Print nCr

**ALGORITHM**

1. Start
2. Read the values of n and r
3. nCr = factorial(n) / ( factorial(r) \* factorial(n-r) );
4. Print nCr
5. Stop

//Function factorial

factorial(num)

begin

1. fact = 1
2. i = 1
3. Repeat step 4 and 5 until i < = num
4. fact = fact \* i
5. i = i+1
6. return(fact)
7. Stop

end

**PROGRAM DESCRIPTION**

n!

---------

r! (n-r)!

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void main()

{

int n,r;

long int nCr;

clrscr();

printf("\n\n\n\t\t\t n! \n");

printf("\t\t\t---------\n");

printf("\t\t\tr! (n-r)!\n");

printf("\n\nEnter the value of n and r : ");

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

nCr = factorial(n) / ( factorial(r) \* factorial(n-r) );

printf("\n\n%dC%d = %ld",n,r,nCr);

getch();

}//End of main function

int factorial(int num)

{

int i;

long fact = 1;

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

fact = fact\* i;

return fact;

}

**OUTPUT**

n!

---------

r! (n-r)!

Enter the value of n and r : 5 2

5C2 = 10

**EXPERIMENT NO : 11**

**polynomial addition**

**AIM** : Program to add two polynomials using function.

**INPUT SPECIFICATION :**

**OUTPUT SPECIFICATION :**

**ALGORITHM**

1. Start
2. Read the maximum power of first and second polynomial ,d1,d2
3. d3 ← sum\_poly(p1,p2,p3)
4. display the two input polynomials and the resultant
5. Stop

sum\_poly(p1[],p2[],res[])

Begin

if(d1 < d2)

Begin

Repeat for I = 0,1,2,………………………d1

res[i] = p1[i] + p2[i]

Repeat for I = d1+1 up to d2

Begin

res[i] ← p2[i]

p1[i] ← 0

End

return d2

End

Else

Begin

Repeat for I = 0,1,2,………………………d2

res[i] = p1[i] + p2[i]

Repeat for I = d2+1 up to d1

Begin

res[i] ← p1[i]

p2[i] ← 0

End

return d1

End

End

**PROGRAM DESCRIPTION**

Get the degrees of two polynomials d1 and d2.

Get the polynomials and add it and display the result.

**PROGRAM**

#include<stdio.h>

#include<conio.h>

int d1,d2;

void main()

{

int i,d3,p1[20],p2[20],res[20];

void read\_poly(int poly[],int n);

void display\_poly(int poly[],int n);

int sum\_poly(int p1[],int p2[],int res[]);

clrscr();

printf("\n\t\tSUM OF TWO POLYNOMIALS\n\n");

printf("\t\t----------------------\n\n");

printf("\n\nEnter the maximum power of 1st and 2nd polynomial : ");

scanf("%d%d",&d1,&d2);

printf("\n\nEnter the coefficients of the 1st polynomial \n");

read\_poly(p1,d1);

printf("\n\nEnter the coefficients of the 2nd polynomial \n");

read\_poly(p2,d2);

//Calls the function sum() to add the 2 polynomials.

d3 = sum\_poly(p1,p2,res);

printf("\n\n\nPolynomial 1 ...................\n\n\n");

display\_poly(p1,d3);

printf("\n\n\nPolynomial 2 ...................\n\n\n");

display\_poly(p2,d3);

printf("\n\n\nSum of the 2 polynomials .......\n\n\n");

display\_poly(res,d3);

getch();

}

void display\_poly(int poly[],int n)

{

int i;

for(i=n;i>=0;i--)

{

printf("%d X^%d ",poly[i],i);

if(i!=0)

printf(" + ");

}

}

void read\_poly(int poly[],int n)

{

int i;

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

{

printf("\nEnter x^%d : ",i);

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

}

}

int sum\_poly(int p1[],int p2[],int res[])

{

int i;

if(d1 < d2)

{

for(i=0;i<=d1;i++)

res[i] = p1[i] + p2[i];

for(i=++d1;i<=d2;i++)

{

res[i] = p2[i];

p1[i] = 0;

}

return(d2);

}

else

{

for(i=0;i<=d2;i++)

res[i] = p1[i] + p2[i];

for(i=++d2;i<=d1;i++)

{

res[i] = p1[i];

p2[i] = 0;

}

return(d1);

}

}

**OUTPUT**

SUM OF TWO POLYNOMIALS

-------------------------------------------

Enter the maximum power of 1st and 2nd polynomial : 2 1

Enter the coefficients of the 1st polynomial

Enter x^0 : 1

Enter x^1 : 2

Enter x^2 : 3

Enter the coefficients of the 2nd polynomial

Enter x^0 : 6

Enter x^1 : 4

Polynomial 1 ...................

3 X^2 + 2 X^1 + 1 X^0

Polynomial 2 ...................

0 X^2 + 4 X^1 + 6 X^0

Sum of the 2 polynomials .......

3 X^2 + 6 X^1 + 7 X^0

**EXPERIMENT NO : 12**

**fibonacci sEries recursively**

**AIM** : Program to generate fibonacci series recursively.

**INPUT SPECIFICATION :** n – the limit of the series

**OUTPUT SPECIFICATION :** Print the Fibonacci series

**ALGORITHM**

1. Start
2. Read the limit of the series ,n
3. Repeat for I = 0,1,2,…………………….up to N

Begin

Call function fib(i)

End

1. Stop

function fib(i)

Begin

if(i=1 or i=0) then

return i

else

return (fib(i-1)+fib(i-2))

End

**PROGRAM DESCRIPTION**

The Fibonacci series is generated using the recursion method.

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,n;

clrscr();

printf("\n\t\tFIBONACCI SERIES USING RECURSION\n");

printf("\t\t--------------------------------\n");

printf("\n\nEnter the limit of fibonacci series : ");

scanf("%d",&n);

printf("\n\nThe fibonacci series is ..............\n\n\n");

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

printf(" %d ",fib(i));

getch();

}//End of main function

int fib(int i)

{

if(i==1 || i==0)

return(i);

else

return(fib(i-1)+fib(i-2));

}

**OUTPUT**

FIBONACCI SERIES USING RECURSION

---------------------------------------------------------

Enter the limit of fibonacci series : 10

The fibonacci series is ..............

0 1 1 2 3 5 8 13 21 34

**EXPERIMENT NO : 13**

**AIM** : To find the details of the employee using structure

**ALGORITHM**

1. Start
2. Create a record employee with the details id ,name,basic salary,netsalary,hra,da and ta
3. Read the employee details
4. Sort the employee details based on employee id
5. Print the details
6. Stop

**PROGRAM DESCRIPTION**

Read the details of employees,calculate net salary and print all details and finally sort employee details with

respect to employee id.

**PROGRAM**

#include<stdio.h>

#include<conio.h>

struct employee

{

int id;

char name[25];

long int bs;

float ns;

struct salary

{

float hra;

float da;

float ta;

}s;

}e[25],tmp;

void main()

{

int i,j,n;

clrscr();

printf("\n\t\tEMPLOYEE DETAILS\n");

printf("\t\t----------------\n\n");

printf("\nEnter number of employees : ");

scanf("%d",&n);

//Details of the employee

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

{

printf("\n\n<< Employee %d >>\n",i+1);

printf("\nEmployee Id : ");

scanf("%d",&e[i].id);

printf("\nName : ");

scanf("%s",&e[i].name);

printf("\nBasic salary : ");

scanf("%ld",&e[i].bs);

}

//Calculates the net salary of the employee

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

{

e[i].s.da = e[i].bs \* 0.15;

e[i].s.ta = e[i].bs \* 0.5;

e[i].s.hra= e[i].bs \* 0.1;

e[i].ns = e[i].bs + e[i].s.da + e[i].s.ta + e[i].s.hra ;

}

//Sort the details based on employee id

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

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

{

if(e[j].id > e[j+1].id)

{

tmp=e[j];

e[j]=e[j+1];

e[j+1]=tmp;

}

}

//Outputs the employee details along with the salary

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

{

printf("\n\n<< Employee %d >>\n",i+1);

printf("\nEmployee Id : %d",e[i].id);

printf("\nName : %s",e[i].name);

printf("\nBasic salary : %ld",e[i].bs);

printf("\nNet salary : %f",e[i].ns);

printf("\n\n");

}

getch();

}

**OUTPUT**

Name : qq

Basic salary : 23

<< Employee 2 >>

Employee Id : 2

Name : q

Basic salary : 12

<< Employee 3 >>

Employee Id : 1

Name : qw

Basic salary : 345

<< Employee 1 >>

Employee Id : 1

Name : qw

Basic salary : 345

Net salary : 603.750000

<< Employee 2 >>

Employee Id : 2

Name : q

Basic salary : 12

Net salary : 21.000000

<< Employee 3 >>

Employee Id : 3

Name : qq

Basic salary : 23

Net salary : 40.250000

**cycle-1v**

**EXPERIMENT NO :14**

**STRING PALINDROME OR NOT**

**AIM :** Program for checking the string is palindrome or not

**INPUT SPECIFICATION :**Give the string

**OUTPUT SPECIFICATION :**Display the string is palindrome or not

**ALGORITHM**

1. Start
2. Read the string
3. Check the length of the string
4. Check each charater of the string from each side.
5. If the string is equal then the string is palindrome else not palindrome
6. stop

**PROGRAM DESCRIPTION**

Program to check the string is palindrome or not

**PROGRAM**

#include<stdio.h>

#include<string.h>

Main()

{

Int i,l,f=0;

Char a[80];

Printf(“\n Enter the string:”);

Gets(a);

L=strlen(a);

For(i=0;i<l;i++)

{

If(a[i]!=a[l-i-1])

{

F=1;

Break;

}

}

If(f==1)

{

Printf(“\n the string is not palindrome “);

}

Else

Printf(“\n string is palindrome”);

}

**OUTPUT**

STRING IS PALINDROME OR NOT

enter the string : malayalam

string is palindrome

enter the string : happy

string is not palindrome

**EXPERIMENT NO : 15**

**sort names using pointers**

**AIM** : Program to sort names using pointers.

**INPUT SPECIFICATION :** N – the number of names to be sorted

A - Array to store the names

**OUTPUT SPECIFICATION :** Display the array A in sorted order.

**ALGORITHM**

1. Start
2. Read how many names N
3. Read the array elements A
4. Cll the function REORDER (n,names)
5. Print “the names in alphabetical order “
6. Stop

function REORDER(N, \*X[])

Begin

Repeat for i = 0,1,……………………upto N-1

Begin

Repeat for j = 0,1,…………………upto n-1-i

Begin

if ( stringcompare( X[ j ] , X[ j+1 ] ) > 0 )

Begin

tmp ← X[j]

X[j] ← X[j+1]

X[j+1] ← tmp

End

End

End

End

**PROGRAM DESCRIPTION**

To sort the names alphabetically in ascending order

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void reorder(int n, char \*x[]);

void main()

{

int i,n;

char \*name[15];

clrscr();

printf("\n\t\tSORT NAMES USING POINTERS\n");

printf("\t\t-------------------------\n\n");

printf("Enter how many names : ");

scanf("%d",&n);

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

name[i]=(char\*)malloc(10\*sizeof(char));

printf("\n\nEnter names \n\n\n");

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

scanf("%s",name[i]);

reorder(n,name);

printf("\nName in alphabetical order \n\n\n");

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

printf("%s\n",name[i]);

getch();

}

void reorder(int n , char \*x[])

{

char \*temp;

int i,j;

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

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

if(strcmp( x[j] , x[j+1] ) > 0 )

{

temp = x[j];

x[j] = x[j+1];

x[j+1] = temp;

}

}

**OUTPUT**

SORT NAMES UING POINTERS

------------------------------------------

Enter how many names : 5

Enter names

Shyam

Bittu

Bunty

Anu

Bittu

Name in alphabetical order

Anu

Bittu

Bittu

Bunty

Shyam

**EXPERIMENT NO : 16**

**swap smallest & largest number**

**AIM** : Program to exchange smallest and largest numbers using pointers.

**INPUT SPECIFICATION :** N - the number of elements in the array

A – the array elements

**OUTPUT SPECIFICATION :** larg - Largest element in the array

small - Smllest element in the array

A - Array elements after sorting

**ALGORITHM**

1. Start
2. Read the number of elements of the array N
3. Read the array elements A
4. \*p ← A
5. larg ← \*p

small ← \*p

1. Repeat for I = 1,2,…………………..upto N

Begin

if( \*(p+i) > larg)

Begin

larg ← \*(p+1)

pl ← i

End

if( \*(p+i) < small)

Begin

small ← \*(p+1)

ps ← i

End

End

1. Print “Largest number ” ,larg

Print “Smallest number ”,small

1. tmp ← \*(p+ps)

\*(p+ps) ← \*(p+ps)

\*(p+ps) ← tmp

1. Print “Array elements after swapping the largest and smallest ”
2. Repeat for I = 0,1,……………….upto N

Begin

Print A[i]

End

1. Stop

**PROGRAM DESCRIPTION**

Exchanges the smallest and largest numbers in an array using pointers

**PROGRAM**

#include<stdio.h>

#include<conio.h>

void swap(int \*x,int \*y);

void main()

{

int a[100],n,i,larg,small,pl=0,ps=0,tmp;

int \*p=a;

clrscr();

printf("\n\t\tEXCHANGE SMALLEST AND LARGEST NUMBER\n");

printf("\t\t------------------------------------\n");

printf("\n\nEnter the number of elements in the array : ");

scanf("%d",&n);

printf("\n\nEnter the array elements : ");

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

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

//Variables for storing the largest and smallest number

larg = \*p;

small= \*p;

/\* Computes the largest and smallest number and also

its array position \*/

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

{

if(\*(p+i) > larg)

{

larg = \*(p+i);

pl = i;

}

if(\*(p+i) < small)

{

small = \*(p+i);

ps = i;

}

}

printf("\nArray elements before swapping : ");

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

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

printf("\n\n\nLargest number = %d\n",larg);

printf("\nSmallest number = %d",small);

//Swaps the smallest and largest number

tmp = \*(p+ps);

\*(p+ps)=\*(p+pl);

\*(p+pl)=tmp;

//Array elements after swapping

printf("\n\n\nArray elements after swapping : ");

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

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

getch();

}

**OUTPUT**

EXCHANGE SMALLEST AND LARGEST NUMBER

---------------------------------------------------------------------

Enter the number of elements in the array : 6

Enter the array elements : 7 3 1 9 4 8

Array elements before swapping : 7 3 1 9 4 8

Largest number = 9

Smallest number = 1

Array elements after swapping : 7 3 9 1 4 8

**EXPERIMENT NO : 17**

**Merge two files**

**AIM** : a program To merge two files

**ALGORITHM**

1. Start
2. Open file1
3. Write data into the file
4. Close the file
5. Open file2
6. Write data into the file
7. Close the file
8. Open file3
9. Open file1
10. Open file2
11. Write data of first file into file3
12. Write data of second file into file3
13. Close all the files
14. Stop

**PROGRAM**

#include<stdio.h>

void main()

{

char s;

FILE \*f1,\*f2,\*f3; //File pointers

clrscr();// Clears the screen

f1=fopen(“first.dat”,”w”); //Opens the 2 files in the write mode

f2=fopen(“second.dat”,”w”);

//Writes data into the 1st file

printf(“Enter first data ‑‑‑ “);

while((s=getchar())!=EOF)

{ putc(s,f1);

}

fclose(f1); //Closes the file

//Writes data into the 1st file

printf(“\nEnter second data ‑‑‑ “);

while((s=getchar())!=EOF)

{ putc(s,f2);

}

fclose(f2); //Closes the file

f1=fopen(“first.dat”,”r”); //Opens the 2 files in the read mode

f2=fopen(“second.dat”,”r”);

f3=fopen(“merge.dat”,”w”);//Opens an output file in the write mode

// Copies the first file into the output file

while(!feof(f1))

{ s=getc(f1);

putc(s,f3);

}

// Appends the second file to the end of the output file

while(!feof(f2))

{ s=getc(f2);

putc(s,f3);

}

//Closes the files

fclose(f1);

fclose(f2);

fclose(f3);

f3=fopen(“merge.dat”,”r”);//Opens the output file for reading

//Prints the merged file

printf(“\nMerged data ‑‑‑\n “);

while(!feof(f3))

{ s=getc(f3);

printf(“%c”,s);

}

fclose(f3);//Closes file

getch();//Accepts a character

}//End of the program

**OUTPUT**

Enter the first data : computer

Enter the second data :science

Merged data

computerscience

**MODEL & VIVA Questions**

1. Write a program to print the nature of roots and calculate the roots of the quadratic equation ax^2+bx+c=0.

2. Write a program to generate Fibonacci series between 10 and 100.

1. To check whether a given number is palindrome or not.
2. Write a program to print Armstrong numbers within a given range.
3. Write a program to check whether the given number is prime or not. If so generate the Fibonacci series up to that number.
4. Write a program to print the numbers in following formats
   1. 1

2 3

4 5 6

7 8 9 10

* 1. 1 c. 1

2 2 2 3 2

3 3 3 3 4 5 4 3

4 4 4 4 4 5 6 7 6 5 4

7. Write a program to generate the series –1,5,-9,13……….to n terms.

4. Write a program to display all prime numbers between 200 and 500.

5. Write a program to convert the given Binary number to its Decimal equivalent.

6. Write a program to print all Armstrong numbers between 10 and 400.

7. Write a program to count the number of spaces, lines, new lines, and tabs in the input.

8. Write a user defined function to raise a number to an exponent.

9. Write a program to compute the mathematical function f (x) =x-x^3/3! +x^5/5! -………

10. Write a program to generate the series 1, 2,6,24,120 to n terms using recursion.

11. Write a program to sort an array of numbers and find the second largest.

1. Write a program to search an element in an array using Binary search method.
2. Write a program to sort words in a line of text in alphabetical order.

12. Write a program to divide a list of numbers into two halves and then sort each half in ascending order

12. Write a recursive function to evaluate S=1+x+x^2+…………….up to x^n.

13. Write a function to evaluate nCr=n! / ((n-r)!\*r!).

14. Write a program to find the sum of diagonal elements of a matrix and its transpose.

15. Write a program to check whether the given string is palindrome or not.

16. Write a program to count the number of occurrences of a given character in a sentence.

17. Write a program to swap 2 characters using a function.

18. Write a program to generate Fibonacci series recursively.

18. Define a structure called DATABASE with members name, code, and basic pay using ‘DATABASE’. Read an array employee with 5 elements .Write a program to read information about all the employees and print them.

19. Write a program to implement a structure called PROGRESS with following members name, rollno, cpmark, lsdmark and sse mark. Read data into the structure and print grade based on the following.

If percentage>80 ‘A’

If percentage >60 ‘B’

If percentage>40 ‘C’

Else ‘FAIL’

20. Write a program to print all prime number leap years from 1901 to 2000.

21. Write a program to condense a number to a single digit by summing its individual digits.(hint:149 becomes 5 (1+4+9=14=1+4=5))

22. Write a program to count the number of vowels present in a given string.

23. Write a program to check if the element of the main diagonal is one.

24. Write a program to find the largest element in the given matrix.

25. Define a structure called ‘automobile’ with the following members model, owner, number and a nested structure year with members manufacture.lastMODdone,nxtMODdone.Write a program to declare a structure car of type automobile and read elements into it and print it.

26. Write a program to sort a single string in alphabetical order.ie.if “adishankara “ is entered the output should be “aaaadhiknrs”.

27. Write a program to concatenate two strings without using the function strcat().

28. Write a program to read consumer number, name and total number of units and generate an electricity bill depending on the following criteria.

If units<=100 rate=Rs.2.50/unit

If units>100 and <=200 rate=Rs.3.50/unit (Rs.2.50 for first100 units)

If units>200 rate=Rs.4.00/unit (Rs.2.50 for first 100,Rs.3.50 for 100 to 200)

29. Write a program to find the row wise sum of m x n matrix.

Write a program to read an array of numbers and split it into two arrays of odd and even numbers.

30. Write a program to write student information stored in a structure into a file and again read the file to display the same. Use fscanf and fprintf function.

31. Write a program to automate a change conversion machine that converts a given amount into appropriate change using denominations of 100,50,5 and 1.eg:if Rs.257 is given, then the output should be

No of Rs.100 notes required: 2

No of Rs.50 notes required: 1

No of Rs.5 noted required: 1

No of Re.1 coin required: 2

32. Read the year and first day of the year from keyboard (say ‘1975’ January is ‘Monday’).Print the calendar of the year.

33. Write a program to read a hexadecimal number and convert it into a decimal number.

34. Read a paragraph. Count number of words. Read a keyword from keyboard and replace it with another word.

35. Write a program to produce the following output using looping statements.

6 6 6 6

4 4 4

2 2

0

36. Write a program to read a string and a key and encode or decode it based on a user choice. Use ‘Offset Cipher Cryptography’ method for encryption and decryption. For eg.the string ‘college team’ encoded with a key value 4 becomes ‘gsppiki xieq’.

37. Write a program to copy the contents of a source file to s destination file. All the

lower-case letters of the source file should appear as upper-case letters in the

destination file.

38. Write a program to merge two arrays into a single array avoiding repetition.

39. Write a program to read a number and convert it into words Eg. 342 should produce Three Hundred Forty Two.

40. WAP to find the saddle point of a matrix.

41. WAP to print all combinations of a string of 4 characters.

42. WAP to check whether the given matrix is triangular. If so, find whether it is upper triangular or lower triangular.

43. WAP to read polling details of 5 candidates (1, 2, 3, 4, 5) and calculate the number of votes obtained by each candidate using array. 0th element will be allotted for invalid polls.

44. WAP to find the trace and norm of a matrix. Trace is defined as the sum of principal diagonal elements. Norm is defined as the square root of the sum of squares of all the elements in a matrix.

45. WAP to input a matrix, determine if it is symmetrical matrix (a matrix is said to be symmetrical when A=A^T)

46. WAP to calculate the total salary of an employee and who gets the highest salary.

DA = 20% of basic

HRA = 30% of basic

47. WAP to extract a substring from the given string.

48. WAP to check whether a given string is palindrome or not.

49. WAP to find the sum of two matrices using pointers.

1. Write a program to reverse the given string using command line arguments.

50. Difference between variable & constant.

51. Size of data types int, double, char.

52. Difference between declarative statement & executable statement

53. Control statements

54. Initialization of elements into 2 X 3 matrix.

55. What is an array?

56. Characteristics of a string.

57. What is sorting?

58. What is a function?

59. Categories of function.

60. Difference between actual & formal parameters.

61. Scope of variables.

62. What is recursion?

63. Difference between recursion & iteration.

64. Difference between call by value & call by reference with example swap.

65. What will be the output of the following program?

main ()

{

printf (“This is C”);

display ();

}

void display ()

{

printf (“Program”);

}

66. What will be the output of the following program?

main ()

{

int i=5;

int a;

a = fun (i);

printf (“%d”,a);

}

fun (int x)

{

if (x>=4)

x = x\*x;

else

x =x\*2;

return (x);

}

67. What is a preprocessor? Which preprocessor is used to define a macro?

68. Difference between a function declaration \* function definition.

69. What is the purpose of return statement?

70. What do you mean by function prototyping?

71. What is a pointer?

72. Array access Pointer Equivalent

Arr [0] ?

Arr [2] ?

Arr[n] ?

73. Difference between Ptr (\*) & ampersand (&) operator.

74. Difference between pre-increment and post increment.

75. Difference between pointer and array.

76. Use of typedef.

77. What is a structure?

78. Difference between structure and union.

79. Difference between structure and array.

80. Explain use of structure with functions.

81. What are the two methods used to access the structure elements using pointers.

82. Write a structure definition having elements : name, roll no, class, sex, height, weight for 50 students.

83. What are files?

84. Types of memory allocations.

85. Dynamic allocation and deallocation functions.

86. Write a simple program to demonstrate the use of dynamic memory allocation fn malloc()

87. Create a structure student with members roll no, name, subject marks. Define its variable for 5 students and initialize its values.

88. Different methods passing structure members to functions.

89. String handling functions.

90. Advantages of arrays.

91. Bitwise operations.

92. Difference between linear search & binary search.

93. Difference between selection sort & bubble sort.

94. Different types of preprocessors.

95. Explain about pointer arithmetic.

96. Functions related to random accessing of files.

97. Basic data types.

98. Input & output statements.

99. Structure of a C Program.

100. Different types of operators.

101. Statements to read a line of text.

102. Statements to read paragraphs.

103. What will be the output of the following?

void main ()

{

start: printf (“WELCOME TO GEC”);

goto start;

}

104. How break and continue statements work in repetitive statement?

105. What are conditional statements?

106. What is the difference between a while and do-while loop.

107. What is the difference between conditional & unconditional goto statements?

108. How a switch-case statement can be differentiated from multiway-if-else statement?

109. What will be the output of the following?

Sum=0;

For (i=1; 1<10; i++)

{

sum+=i;

}

110. Steps to read a list of strings.

111. What are subscripts?

112. What is macro?

113. Different types of macros.

114. How pointers can be used with the string of characters?

115. int a, b, \*ptr;

ptr=&a;

a=9;

b=\*ptr;

a++;

What is the value of a, b, \*ptr?

116. WAP segment to exchange a string from one variable to another using pointer.

117. What will be the o/p of the following program segment?

void main()

{

int a=2;

switch (a)

{

case 1:

printf (“One”);

break;

case 2:

printf(“Two”);

case 3:

printf(“Three”);

default:

printf(“Invalid”);

}

}

118. Purpose of header files.

119. Explain about type casting.

120. What is compiler?

121. What is Size, range.

122. What are the parts of loop expression in ‘for’ loop?

123. What are the differences between object oriented programming & structured programming?

124. What is command line argument?

125. What are the different modes to open a file?

126. WAP segment to read all character from a file.

**REFERENCES**

1. Programming with C - Byron .S. Gottfied, Tata Megraw Hill

2. Computer Programming in C - Kerninghan &Ritachic, PHI

3. Let us C - Yeshwanth Karetkar

4. Pointers in C - Yeshwanth Karetkar

5. Programming with C - Schaum’s Outline Series