

**Department of Computer Science and Engineering**

**“Jnana Sangama”, VTU-Campus, Belagavi-590018**

**Course: Principles of Programming Using C Code: BPOPS103/203**

**LABORATORY**

**MANUAL**

**Academic Year: 2023-24**

Prepared by:

Mahesh Karagaonkar

Keerti Mulik

Programming Assignments:

1 Simulation of a Simple Calculator.

2 Compute the roots of a quadratic equation by accepting the coefficients. Print appropriate messages.

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

4. Write a C Program to display the following by reading the number of rows as input, 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 --------------------------- nth row

5 Implement Binary Search on Integers.

6 Implement Matrix multiplication and validate the rules of multiplication.

7 Compute sin(x)/cos(x) using Taylor series approximation. Compare your result with the built-in library function. Print both the results with appropriate inferences.

8 Sort the given set of N numbers using Bubble sort.

9 Write functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.

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

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

12. Write a C program to copy a text file to another, read both the input file name and target file name

1.Simulation of a Simple Calculator.

Algorithm:

1. Start

2. Initialize variables:

- num1 (operand 1)

- num2 (operand 2)

- operator (operation to perform)

- result (result of the operation)

3. Display a menu to the user with options:

a. Enter operand 1

b. Enter operand 2

c. Choose operation (+, -, \*, /)

d. Calculate

e. Exit

4. Repeat the following steps until the user chooses to exit:

a. Prompt the user to choose an option from the menu.

b. If the user chooses option 'a':

i. Prompt for and input operand 1.

ii. Go back to step 3.

c. If the user chooses option 'b':

i. Prompt for and input operand 2.

ii. Go back to step 3.

d. If the user chooses option 'c':

i. Prompt for and input the operation (+, -, \*, /).

ii. Go back to step 3.

e. If the user chooses option 'd':

i. Perform the selected operation based on the entered operands and operator.

ii. Display the result.

iii. Go back to step 3.

f. If the user chooses option 'e':

i. Exit the program.

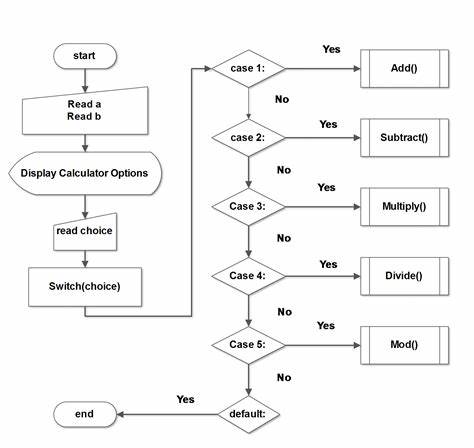
g. If the user enters an invalid option:

i. Display an error message.

ii. Go back to step 3.

5. End

Flowchart:



Program :

#include <stdio.h>

int main() {

float num1, num2, result;

char operator;

// Input

printf("Enter first number: ");

scanf("%f", &num1);

printf("Enter operator (+, -, \*, /): ");

scanf(" %c", &operator);

printf("Enter second number: ");

scanf("%f", &num2);

// Calculation

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

// Check if the denominator is not zero

if (num2 != 0):

{

result = num1 / num2;

}

else :

{

printf("Error: Division by zero is undefined.\n");

return 1; // Exit the program with a

}

break;

default:

printf("Error: Invalid operator.\n");

return 1; // Exit the program with an error code

}

// Output

printf("Result: %.2f\n", result);

return 0;

}

OUTPUT 1:

Enter first number: 10

Enter operator (+, -, \*, /): \*

Enter second number: 5

Result: 50.00

OUTPUT 2:

Enter first number: 10

Enter operator (+, -, \*, /): +

Enter second number: 5

Result: 15.00

2.QUADRATIC EQUATION:

AIM: To find the roots of the quadratic equation (ax2+bx+c=0) with different possible input

values for a, b and c.

ALGORITHM :

ALGM: Quadratic Equation [This algorithm takes three coefficients as input and computes the roots]

Steps:

1. [Initialize] Start

2. [Input coefficients of quadratic equation]

read a ,b, c

3. [Check for valid coefficients]

If a =0 and b= 0 then

print “Roots cannot be determined”

[Check for linear equation]

else a=0then

root1 ← (-c/b)

print “Linear equation”,root1

goto step 5

4. [Compute discriminate value]

disc ← b\*b-4\*a\*c

5. [Based on discriminate value, classify and calculate all possible roots and print them]

5.1 [If discriminate value is 0, roots are real & equal.]

if disc=0 then

root1← root2 ← (-b/2\*a)

` print “Real & equal roots”, root1, root2

5.2 [ If discriminate value is >0, roots are real & distinct.]

else if disc>0then

root1← (-b+√disc)/(2\*a)

root2 ← (-b-√disc)/(2\*a)

print “Real & distinct roots”, root1, root2

5.3 [ If discriminate value is <0, roots are imaginary.]

else

real ← -b/(2\*a)

imag ← √(fabs(disc))/(2\*a)

root1 ← (real) + i (imag)

root2 ← (real) - i (imag)

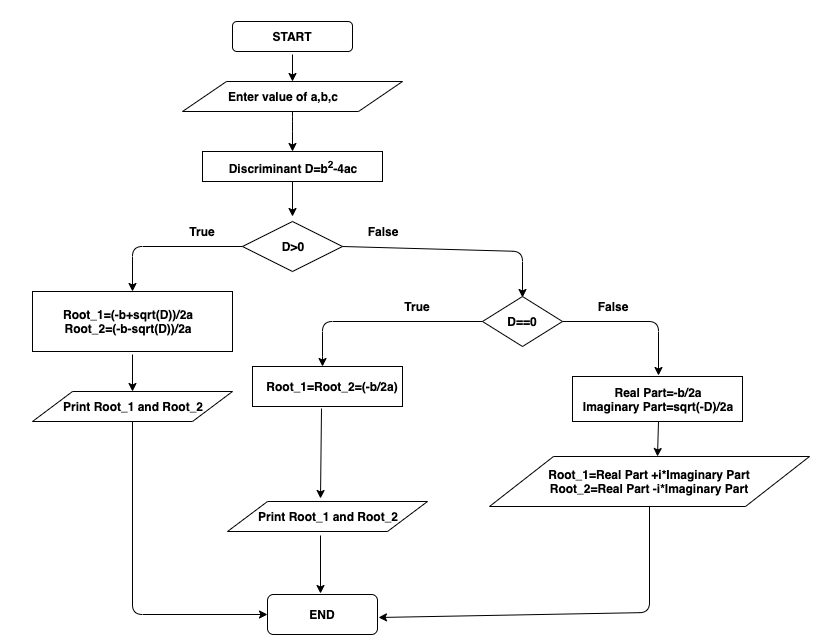
print “Imaginary roots”, root1, root2

endif

endif

6. [Finished] End

Flowchart:



PROGRAM :

/\* Program to calculate all possible roots of a quadratic equation \*/

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main()

{

float a, b, c, disc;

float root1,root2,real,imag;

clrscr();

printf("Enter a,b,c values\n");

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

if( (a == 0) && (b == 0) &&(c==0))

{

printf("Invalid coefficients\n");

printf(" Try Again with valid inputs !!!!\n");

getch();

}

disc = b\*b - 4\*a\*c;

if(disc == 0)

{

printf("The roots are real and equal\n");

root1 = root2 = -b/(2\*a);

printf("Root1 = %.3f \nRoot2 = %.3f", root1,root2);

}

else if(disc>0)

{

printf("The roots are Real and Distinct\n");

root1 = (-b+sqrt(disc)) / (2\*a);

root2 = (-b-sqrt(disc)) / (2\*a);

printf("Root1 = %.3f \nRoot2 = %.3f",root1,root2);

}

else

{

printf("The roots are Real and Imaginary\n");

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

imag = sqrt(fabs(disc)) / (2\*a);//fabs() returns only numberignoring sign

printf("Root1 = %.3f + i %.3f \n",real,imag);

printf("Root2 = %.3f - i %.3f",real,imag);

}

getch();

OUTPUT:

1. Enter a, b, c values

0 0 1

Invalid coefficients

Try Again with valid inputs!!!!

2. Enter a, b, c values

1 2 3

The roots are Real and Imaginary

Root1 = -1.000 + I 1.414

Root2 = -1.000 – I 1.414

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

Algorithm:

Input: Customer name and unit consumed

Output: Customer name, units consumed and total amount to be paid

Step 1: Start

Step 2: Read the name of customer and the unit consumed by the customer.

Step 3: Check if the unit consumed is greater than 1 and less than 200,if true goto step 4 else goto step 5.

Step 4: Compute: amt=100+(0.8\*units).

Step 5: if unit is greater than 200 and less than 300,if true goto step 6 else goto step 7

Step 6: Compute amt=100+(200\*0.8)+((units-200)\*0.9)

Step 7:Compute amt=100+(200\*0.8)+(100\*0.9)+((units-300)\*1), then goto step 8

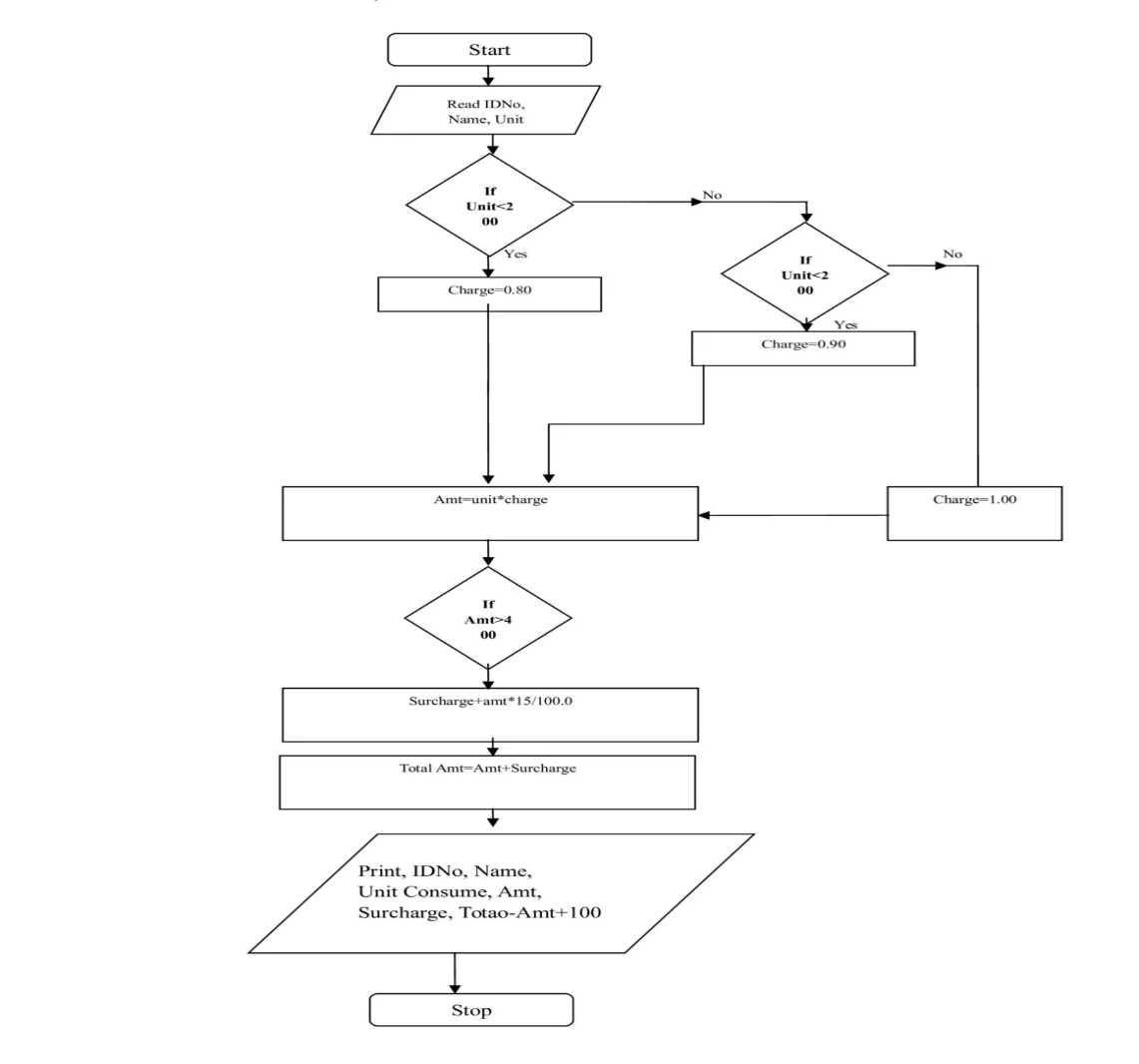
Step 8: Check if the amt is less than or equal to 400, if true goto step 9 otherwise goto step 10.

Step 9: Print the amount charged and goto step 11.

Step 10: compute amt=amt\*1.15,and print the amount charged.

Step 11: Stop

Flowchart:



Program

#include <stdio.h>

void main()

{

char name[10];

float unit, amt;

printf("Enter your name and unit Consumed:");

scanf("%s %f",name,&unit);

if(unit<=200)

amt=unit\*0.80+100;

else if((unit>200)&&(unit<=300))

amt=200\*0.80+((unit-200)\*0.90)+100;

else

amt=200\*0.80+100\*0.90+((unit-300)\*1)+100;

if(amt>400)

amt=1.15\*amt;

printf("Name: %s\n Unit=%f \n charge=%f ",name,unit,amt);

}

Output

First Run

Enter your name and unit Consumed: Siri 52

Name: Siri

Unit=52

charge=141.600000

Second Run

Enter your name and unit Consumed: Rajesh 460

Name: Rajesh

Unit=460

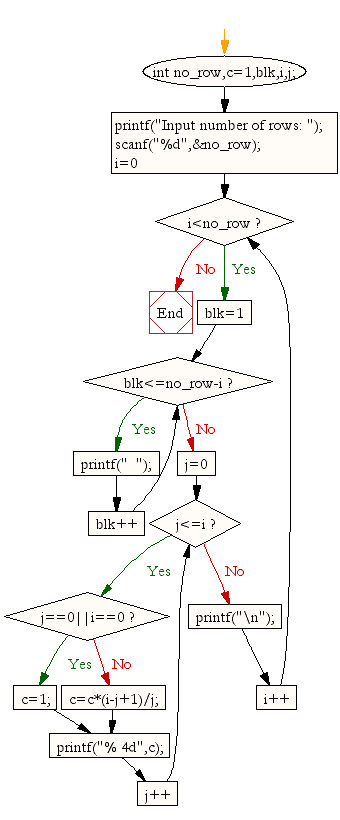
charge=586.500000

4. Write a C Program to display the following by reading the number of rows as input, 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 --------------------------- nth row

Algorithm:

1. Read the number of rows (n) as input.
2. Use a loop to iterate through each row from 1 to n.
3. For each row, print decreasing values from the current row number to 1.
4. For each row, print increasing values from 2 to the current row number.
5. Move to the next line after completing each row.

Flowchart:



Program:

#include <stdio.h>

void printPattern(int n) {

// Loop through each row

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

// Print decreasing values

for (int j = i; j >= 1; j--) {

printf("%d ", j);

}

// Print increasing values excluding the first element

for (int j = 2; j <= i; j++) {

printf("%d ", j);

}

printf("\n");

}

}

int main() {

int rows;

// Input the number of rows

printf("Enter the number of rows: ");

scanf("%d", &rows);

// Display the pattern

printPattern(rows);

return 0;

}

Output:

Enter the number of rows: 5

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

5.Implement Binary Search on Integers.

AIM: To search a name in list of names using Binary Searching Technique

ALGORITHM:

ALGM: Binary search

OPTIMAL-BST(p, q, n)

1 for i = 1 to n + 1

2 do e[i, i - 1] = qi-1

3 w[i, i - 1] = qi-1

4 for l =1 to n

5 do for i = 1 to n - l + 1

6 do j = i + l - 1

7 e[i, j] = âˆž

8 w[i, j] = w[i, j - 1] + pj + qj

9 for r = i to j

10 do t = e[i, r - 1] + e[r + 1, j] + w[i, j]

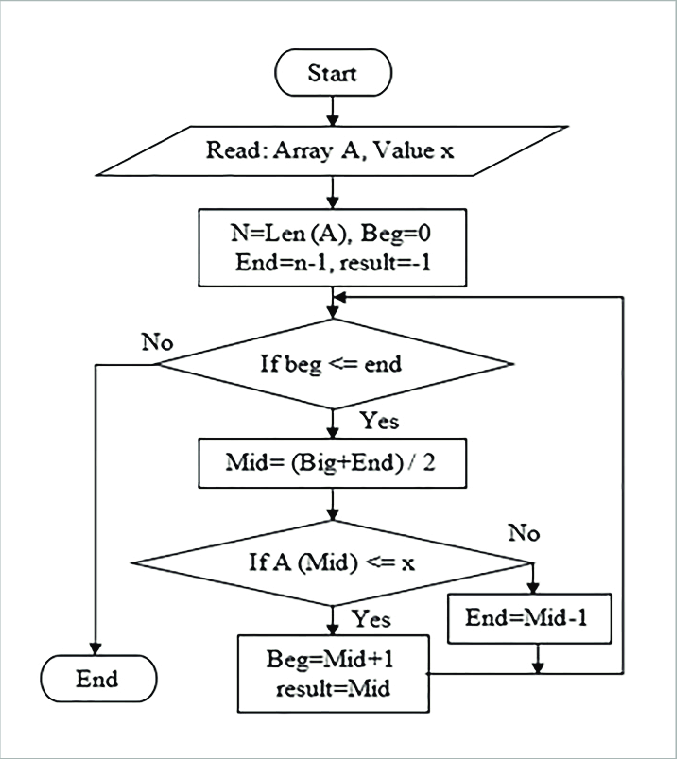
11 if t < e[i, j]

12 then e[i, j] = t

13 root[i, j] = r

14 return e and root

Flowchart:



PROGRAM :

/\* progrm to search an name using binary search \*/

#include<stdio.h>

#include<string.h>

#include<conio.h>

int main()

{

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

int n, i, low, high, mid, found=0;

clrscr();

printf("Enter the number of names to read, n=");

scanf("%d", &n);

printf("Enter the names in ascending order\n");

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

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

printf("Enter the name to be search:");

scanf("%s", key);

low=0;

high=n-1;

while(low<=high && !found)

{

mid=(low + high)/2;

if(strcmp(name[mid],key)==0)

found=1;

else if(strcmp(name[mid],key)<0)

low=mid+1;

else

high=mid-1;

}

if(found == 1)

printf("Name found in position : %d",mid+1);

else

printf("Name not found");

getch();

}

OUTPUT:

1. Enter the number of names to read, n= 5

Enter the names in ascending order

Amar

Chethan

Girish

Manoj

Yadu

Enter the name to be search:

Chethan

Name found in position :2

2. Enter the number of names to read, n= 5

Enter the names in ascending order

Girish

Manoj

Yadu

Amar

Chethan

Enter the name to be search:

Kiran

Name not found

6.Implement Matrix multiplication and validate the rules of multiplication.

AIM: To read two matrices A(m x n )and B(p x q) and Compute the product A and B.

ALGORITHM:

ALGM: matrix multiplication

Step 1: GET THE MATRIX SIZE OF a

Input the size of matrix a and read the values of m and n.

Step 2: GET THE MATRIX VALUE OF a

For i=0 to n-1

For j=0 to n-1

Read a[i][j]

End For

End For

Step 3: GET THE MATRIX SIZE OF b

Input the size of matrix b and read the values of p and q.

Step 4: GET THE MATRIX VALUE OF b

For i=0 to p-1

For j=0 to q-1

Read b[i][j]

End For

End For

STEP :5 MULTIPLICATION OF MATRICES NOT POSSIBLE

If(n!=p)

Print(“multiplication is not possible”)

Stop

End if

STEP 6: MULTIPLICATION OF MATRICES IS POSSIBLE

For i=0 to m-1

For j=0 to q-1

Sum=0

For k=0 to n-1

Sum=Sum+a[i][k]\*b[k][j]

End for

End for

Step 7: DISPLAY RESULT

For i=0 to m-1

For j=0 to q-1

Print c[i][j]

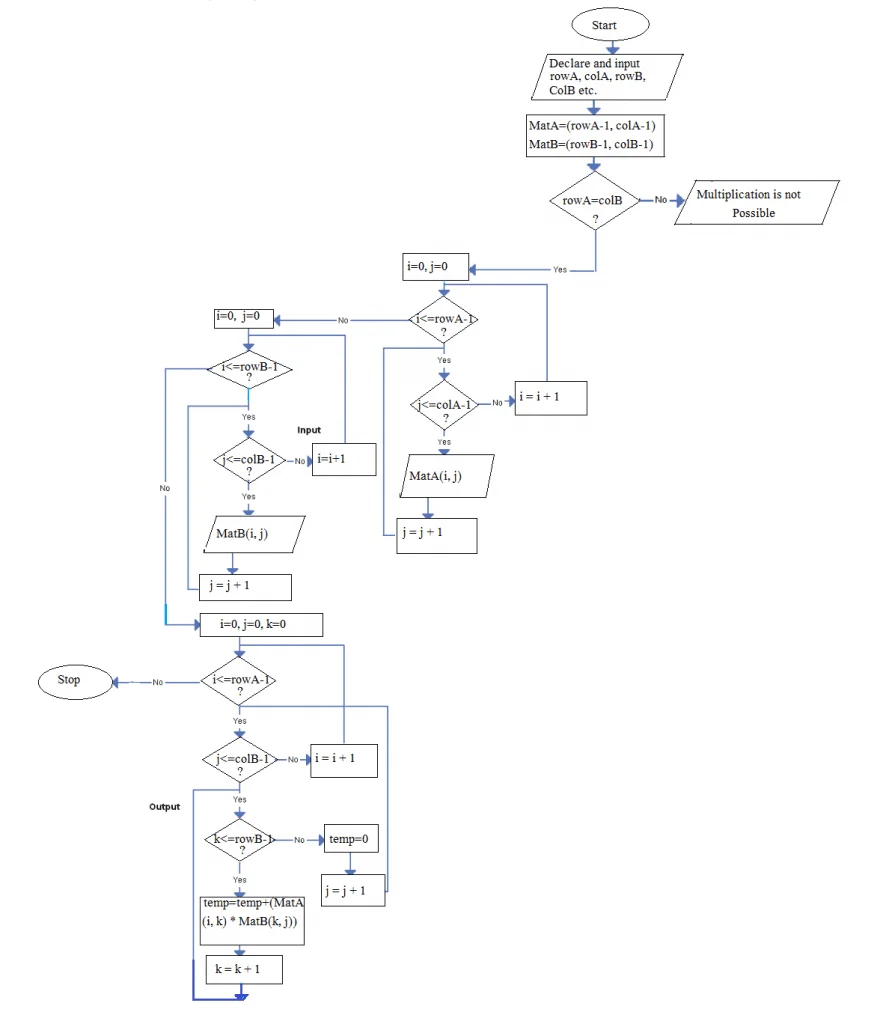
End for

End for

Step 8: STOP

stop

Flowchat:



PROGRAM :

#include<stdio.h>

#include<conio.h>

int main()

{

int a[5][5],b[5][5],c[5][5],m,n,p,q,i,j,k;

clrscr();

printf("Enter the size of first matrix\n");

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

printf("Enter the size of second matrix\n");

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

if(n!=p)

printf(“Matrix multiplication is not possible”);

else

{

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

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

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

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

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

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

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

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

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

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

{

c[i][j]=0;

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

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

}

printf("\n A- matrix is\n");

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

{

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

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

printf("\n");

}

printf("\n B- matrix is\n");

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

{

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

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

printf("\n");

}

printf("The product of two matrix is\n");

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

{

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

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

printf("\n");

}

}

getch();

}

OUTPUT:

1. Enter the size of first matrix

2 3

Enter the size of second matrix

3 2

Enter the elements of first matrix

1 2 3 4 5 6

Enter the elements of the second matrix

1 2 3 4 5 6

A- matrix is

1 2 3

4 5 6

B- matrix is

1 2

3 4

5 6

The product of two matrix is

22 28

49 64

7 Compute sin(x)/cos(x) using Taylor series approximation. Compare your result with the built-in library function. Print both the results with appropriate inferences.

ALGORITHM: TAYLOR SERIES

Step 1: Start

Step 2: Read angle x

Step 3: Assign y=x

Step 4: Compute the value of x in radians:

x = (3.1412/180.0) \* degree

Step 5: Initialise:

i. sum = x

ii. t= x

iii. i=1

Step 6: Compute:

i. i = i + 2

ii. t = ( - t \* x \* x ) / ( (i-1) \* i )

iii. Sum = sum + t

Step 7: Check if the absolute value term is greater than 0.00005. If true goto Step 6 else goto

Step 8

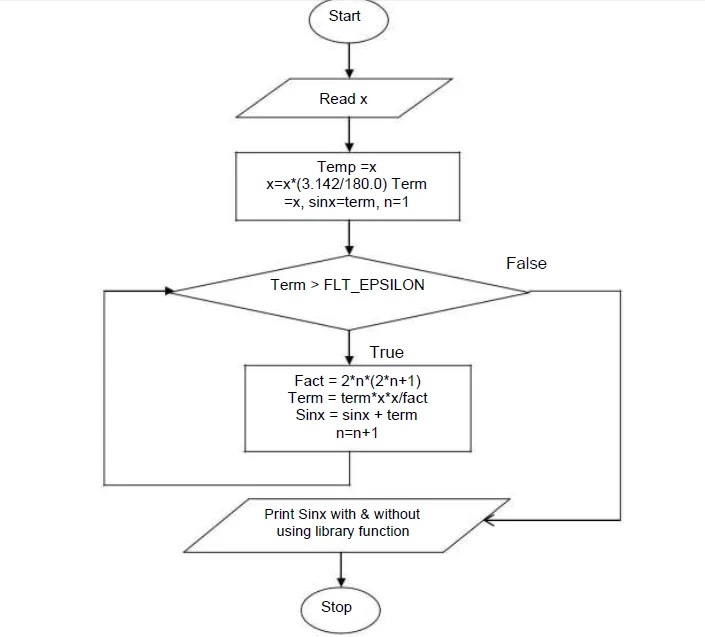
Step 8 : Display sine using iteration : sum

Step 9 : Assign res=sin(x);

Step 10 : Display sine using library function: sin(x)

Step 11: Stop

Flowchart:



Program:

#include<stdio.h>

#include<math.h>

int main( )

{

int i;

float x,t,sum,sum1,y;

printf("Enter the angle\n");

scanf("%f",&x);

y=x;

x=3.1428\*(x/180.0);

sum=x;

t=x;

i=1;

do

{

i=i+2;

t=(-t\*x\*x)/((i-1)\*i);

sum=sum+t;

}while(fabs(t)>0.00005);

printf("sin(%f) using taylor series=%f\n",y,sum);

sum1=sin(x);

printf("Using inbuilt function sin(%f)=%f",y,sum1);

}

Output:

Enter the angle 30

sin(30.000000) using taylor series=0.500000

Using inbuilt function sin(30.000000)=0.500000

8.Sort the given set of N numbers using Bubble sort.

ALGORITHM: BUBBLE SORT

Input: A unsorted array of n numbers

Output: sorted array

Step 1: START

Step 2: Read unsorted array n elements in to a[i], where i is the index value.

Step 3: Initialize index i=0.

Step 4: Check if i is less than n. if true, goto step 5. Otherwise goto step output array.

Step 5: initialize index j to zero.

Step 6: check if j is less than (n-i-1). If true goto step 7. Otherwise increment j and goto step 4.

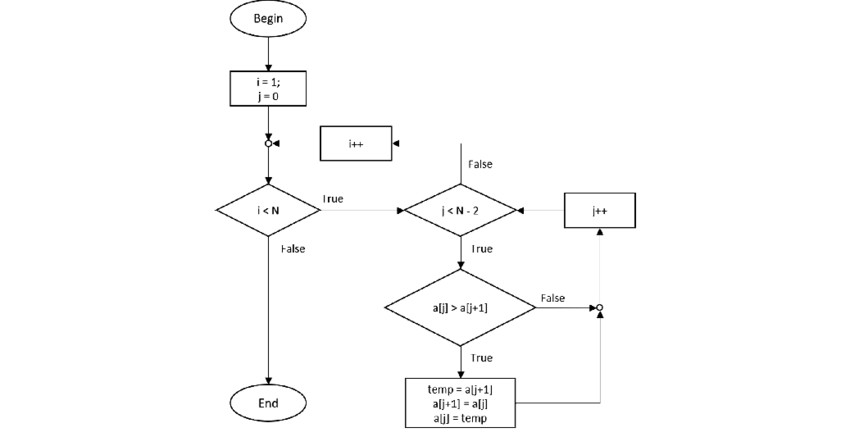
Step 7: inside the for loop check if a[j] is greater than a[j+1](i.e., adjacent elements are compared). If

true swap elements using temporary variables. Otherwise goto step 6.

Step 8: Output the sorted array elements using for loop.

Step 9: Stop

Flowchat:



Program :

#include<stdio.h>

int main()

{

int i,j,n,temp;

int a[20];

printf("enter the value of n");

scanf("%d",&n);

printf("Enter the numbers in unsorted order:\n");

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

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

// bubble sort logic

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

{

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

{

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

{

temp=a[j];

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

a[j+1]=temp;

}

}

}

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

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

{

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

}

}

Output:

Enter the value of n

5

Enter the numbers one by one in unsorted order:

20

10

30

50

40

The sorted array is

10

20

30

40

50

9 Write functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.

ALGORITHM: STRING OPERATIONS

Step 1: Start

Step 2: press 1-compare 2-concatenate 3-length of string, if press 1 goto step 3, if press 2 goto step 4,

if press 3 goto step 5,

Step 3: Read String1 and String 2 and Comparison function by using strcmp built in function is used.

if res=0 then print both strings are equal, else print strings are not equal and goto step 4.

Step 4: Read String1 and String 2 and find concatenation of two strings using string handling

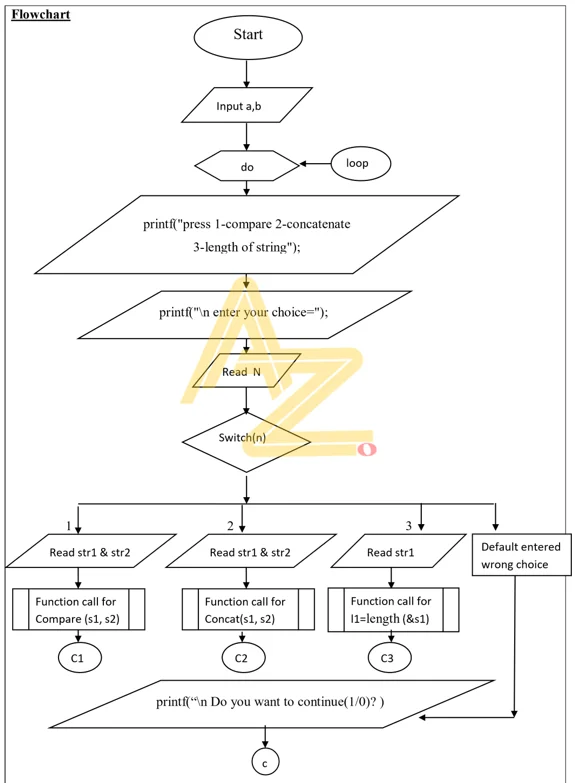
function strcat( ) and display string and return back to main function.

Step 5: Read String1and call function to find the length of string by calling function length(\*string)

Step 6: if digit=1 then goto step 2 otherwise goto step 7

Step 7: stop

Flowchart:



Program :

#include<stdio.h>

#include<string.h>

void compare(char [ ],char [ ]); .

void concat(char [ ],char [ ]);

void length(char \*[ ]);

void main( )

{

int n,digit;

char str1[10],str2[10];

do

{

printf("press 1-compare 2-concatenate 3-length of string");

printf("\n enter your choice=");

scanf("%d",&n);

switch(n)

{

case 1:printf("enter first string=");

scanf("%s",str1);

printf("enter second string=");

scanf("%s",str2);

compare(str1,str2);

break;

case 2: printf(“enter first string=");

scanf("%s",str1);

printf("enter second string=");

scanf("%s",str2);

concat(str1,str2);

break;

case 3:printf("enter string=");

scanf("%s",str1);

length(&str1);

break;

default: printf("wrong choice");

break;

}

printf("\n Do you want to continue(1/0)? ");

scanf("%d", &digit);

}while(digit==1);

}

void compare(char str1[ ],char str2[ ])

{

int i;

i=strcmp(str1,str2);

if(i==0)

printf("strings are equal\n ");

else

printf("string are not equal\n");

}

void concat(char str1[ ],char str2[ ])

{

strcat(str1,str2);

printf("concatenate string=%s",str1);

}

void length(char \*str1[ ])

{

int len;

len=strlen(str1);

printf("the length of string=%d",len);

}

Output:

press 1-compare 2-concatenate 3-length of string

enter your choice=1

enter first string=ram

enter second string=Ram

string are not equal

Do you want to continue(1/0)? 1

press 1-compare 2-concatenate 3-length of string

enter your choice=2

enter first string=RAM

enter second string=kumar

concatenate string=RAMkumar

Do you want to continue(1/0)? 1

press 1-compare 2-concatenate 3-length of string

enter your choice=3

enter string=ram the length of string=3

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

ALGORITHM: STUDENT DETAILS

Input: Student details such as student id, name and marks

Output: To print the details of those students scoring above and below average

Step 1: START

Step 2: Read the number of students

Step 3: For each student, read the student id, name and marks for all subjects.

Step 4: Calculate the average of the marks and store in the avg field.

Step 5: Print results.

Step 6: Initialise loop

Step 7: Read the average of each student

Step 8: Check if avg>35.00

Step 9: If yes print result else go to next iteration

Step 10: Initialise loop

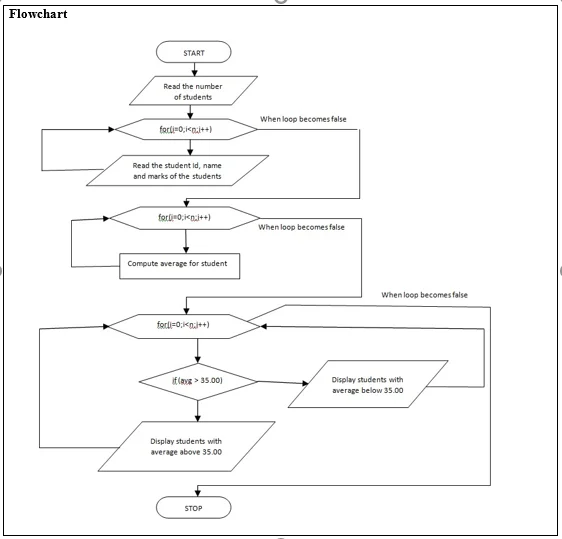
Step 11: Read average of each student

Step 12: Check if avg<35.00

Step 13: If yes print result else go to next iteration

Step 14:STOP

Flowchart:



Program 10

#include<stdio.h>

struct student

{

char usn[10];

char name[10];

float m1,m2,m3;

float avg,total;

};

void main()

{

struct student s[20];

int n,i;

float tavg,sum=0.0;

printf("Enter the number of student=");

scanf("%d",&n);

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

{

printf("Enter the detail of %d students\n",i+1);

printf("\n Enter USN=");

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

printf("\n Enter Name=");

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

printf("Enter the three subject score\n");

scanf("%f%f%f",&s[i].m1,&s[i].m2,&s[i].m3);

s[i].total=s[i].m1+s[i].m2+s[i].m3;

s[i].avg=s[i].total/3;

}

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

{

if(s[i].avg>=35)

printf("\n %s has scored above the average marks",s[i].name);

else

printf("\n %s has scored below the average marks",s[i].name);

}

}

Output:

Enter the number of student=2

Enter the detail of 1 students

Enter USN=101

Enter Name=Ram

Enter the three subject score 10 21 15

Enter the detail of 2 students

Enter USN=102

Enter Name=Kumar

Enter the three subject score 11 9 10

Ram has scored above the average marks

Kumar has scored below the average marks

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

ALGORITHM: SUM, MEAN, VARIANCE

Input: read numbers to compute sum, mean, variance and deviation

Output: results sum, mean, variance and deviation

Step1: START

Step2:Read n

Step3:for each value of n read x

Step4: initialize sum=0, i=0

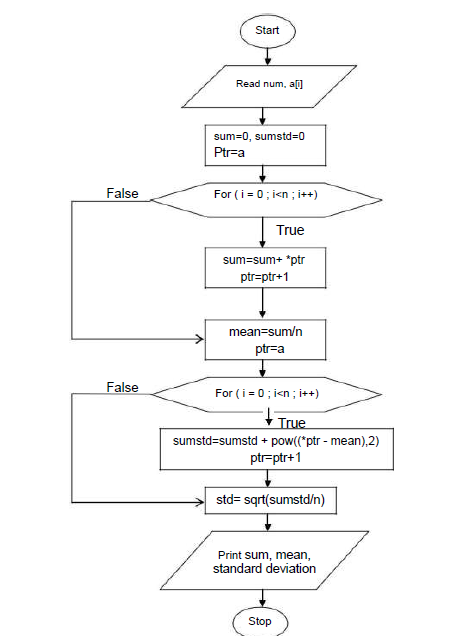
Step5: for each value of n and i, Compute sum using sum=sum+(\*(x+i)-mean)\*(\*(x+i)-mean)

Step 6: using sum value compute variance=sum/n and deviation=sqrt(variance)

Step 7: display mean, variance, deviation

Step 8: Stop

Flowchart:



Program:

#include<stdio.h>

#include<math.h>

int main()

{

int n , i;

float x[20],sum,mean;

float variance , deviation;

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

scanf("%d",&n);

printf("enter %d real values \n",n);

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

{

scanf("%f",(x+i));

}

sum=0;

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

{

sum= sum+\*(x+i);

}

printf("sum=%f\n",sum);

mean=sum/n;

sum=0;

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

{

sum=sum+(\*(x+i)-mean)\*(\*(x+i)-mean);

}

variance=sum/n;

deviation=sqrt(variance);

printf("mean(Average)=%f\n",mean);

printf("variance=%f\n",variance);

printf("standard deviation=%f\n",deviation);

}

Output:

Entre the value of n

5

Enter 5 real values

3

7

23

1

4

sum=38.000000

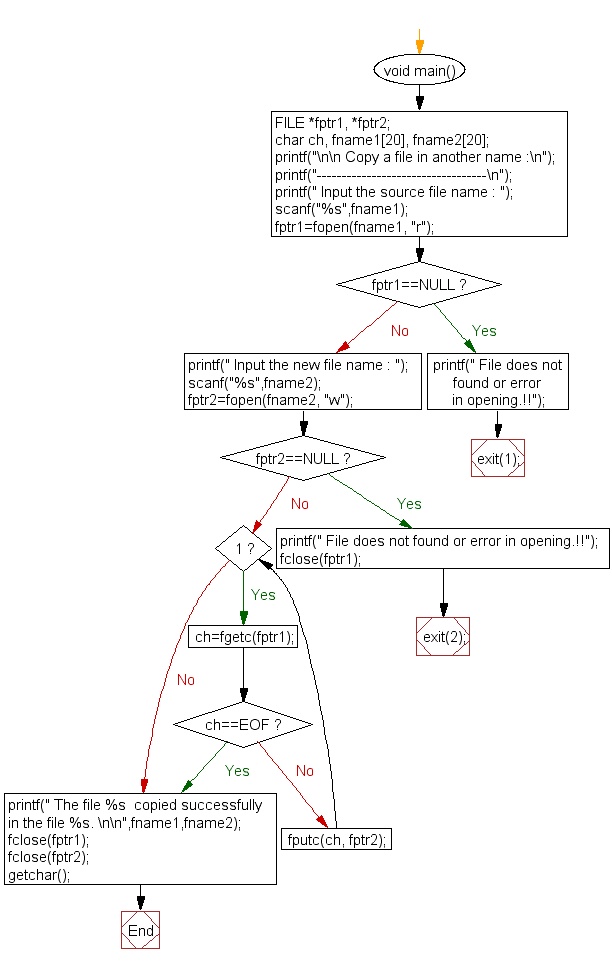
mean (Average)=7.600000

variance=63.039997

standard deviation=7.939773

12. Write a C program to copy a text file to another, read both the input file name and target file name

Flowchart:



Algorthim:

STEP 1: Create a New Text Document and Rename it as "abc.txt"

STEP 2: Write Some Content in that file and Save.

STEP 3: Now Compile the C Program and RUN it.

That's it.

Now a new file will be created with a name "xyz.c" and all the content would copy from abc.txt to xyz.txt.

Program:

#include<stdio.h>

#include<stdlib.h>int main(void)

{

FILE \*fp1,\*fp2;

int ch;

char fname1[100], fname2[100];

printf("\nEnter File name to be copied\n");

scanf("%s",fname1);

fp1 = fopen(fname1,"r");

if(fp1 == NULL)

{

printf("\nInput File %s doesn’t exist\n", fname1);

exit(0);

}

printf("\nEnter target File name\n");

scanf("%s",fname2);

fp2 = fopen(fname2,"w");

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

{

fputc(ch,fp2);

}

printf("\nFile %s successfully created\n",fname2);

fclose(fp1);

fclose(fp2);

return 0;

}

Output

=================================

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter File name to be copied

out9.c

Enter target File name

out99.c

File out99.c successfully created

===============================================

Enter File name to be copied

secret.txt

Input File secret.txt doesn’t exist

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/