Commands to run C++ program
g++ .\trial.cpp
 ./a.out   (on linux vs code)

On vscodewindows:
g++ .\trial.cpp
./a.exe

python3 filename.py

c++ in linux text editor
make filename
./filename

Assign 1- Trees DFS, BFS

DFS- post order

BFS- level order

```
#include <bits/stdc++.h>
using namespace std;

class node
{
    //keeping all variables public
    public:
    //value
    int data;
    //pointers
    node* left;
    node* right;
    //constructor
    node()
    {
        left = right = NULL;
    }
};

class tree
{
    //keeping root as private
    node *root;
    public:
    //constructor
    tree()
    {
        root = NULL;
    }
    node* getRoot()
    {
        return root;
    }
```

```
    bool isEmpty()
    {
        return (root == NULL);
    }
    void insert();
    void DFS(node* root);
    void BFS(node* root);
};
void tree :: insert()
{
    node*p = new node();
    node* curr = root;
    cout<<"Enter the value - "<<endl;
    int val;
    cin>>val;
    p->data = val;
    p->left = p->right = NULL;
    if(isEmpty())
    {
        root = p;
    }
    else
    {
        int flag = 0;
        do
        {
            if(p->data>curr->data)
            {
                if(curr->right != NULL)
                {
                    curr = curr->right;
                }
                else
                {
                    curr->right = p;
                    flag = 1;
                }
            }
            else if(p->data<curr->data)
            {
                if(curr->left != NULL)
                {
                    curr = curr->left;
                }
                else
                {
                    curr->left = p;
                    flag = 1;
                }
            }
            else
            {
                cout<<"Duplicate data not allowed"<<endl;
```

```
                flag = 1;
            }
        }while(flag!=1);
    }
}

void tree::BFS(node*root)
{
    if(isEmpty())
    {
        cout<<"No values added"<<endl;
        return;
    }
    //create queue
    queue <node*>q;
    //add root
    q.push(root);
    while(!q.empty())
    {
        node*curr = q.front();
        cout<<curr->data<<" ";
        q.pop();
        //push left child
        if(curr->left != NULL)
        {
            q.push(curr->left);
        }
        //push right child
        if(curr->right != NULL)
        {
            q.push(curr->right);
        }
    }
    cout<<endl;
}

void tree::DFS(node*root)
{
    if(root == NULL)
    {
        return;
    }
    else
    {
        DFS(root->left);
        DFS(root->right);
        cout<<root->data<<" ";
    }
    cout<<endl;
}

int main()
{
```

```
    tree t1;
    char ans;
    do
    {
      int choice;
      cout<<"1.Insert Nodes\n2.Breadth First Traversal\n3.Depth First
Traversal"<<endl;
      cin>>choice;
      switch(choice)
      {
          case 1:
          t1.insert();
          break;
          case 2:
          t1.BFS(t1.getRoot());
          break;
          case 3:
          t1.DFS(t1.getRoot());
          break;
          default:
          cout<<"Wrong choice"<<endl;
      }
      cout<<"Enter menu again(y/n"<<endl;
      cin>>ans;
    }while(ans=='y');
}

Complexity for both DFS & BFS - O(n)
BFS - queue (FIFO)
DFS - Stack (LIFO)

//If bits/stdc++.h doesn't work then use iostream,queue

#Check dfs, bfs sequence once
#Check if we can improve complexity
#Check alternative approaches
```

Assign 2- A* Algorithm

So simply speaking we have a start node and end node and we have to write a code to go from start to end but there is one catch - we have to calculate heuristic at every step for node and decide what to do
To calculate the heuristic -> distance of the node from start + distance of node from end
Har baar we can't know the distance from start for each node coz apan he sab ek graph wale path par kar rahe so we simply do distance of its parent node from start +1

Pseudo Code -

- We take start and end node as input from user
- Apan __init__ mein har node ko parent aur position yeh do attributes hein
- Parent -> we'll need once we have found end node toh we backtrack using it to get path
- Position -> (x,y) to get position
- We initialize 2 empty lists -> open list & closed -> open list means unprocessed nodes and closed list means processed nodes

- Ab assuming that prob is correct, we'll have to start from start node and keep on going till we reach the end node
- So pehle bina kuch soche samjhe start node ko open list mein add kardo
- We then loop till open list is empty - matlab koi unprocessed nahi matlab you got to end node
    - We then traverse through open list and find node with least heuristic
    - Usse open list se pop karte hein aur closed mein add karte hein
    - Now we first check ki if its the end node
        - Agar hoga toh baat khatam
        - We only have to loop bak until we get parent is null and add path to list
        - Abhi path is end to start so we reverse it to get it from start to end
    - Agar end node nahi toh we need to go ahead, so we add its neighour nodes to open list - par apan directly nahi add kar sakte, kaafi checking karni padegi
    - Sabse dekho ki agar wo neighbor range mein hein na, coz corner nodes ko khali 3 neighbors hote instead of 8
    - Woh hone ke baad dekho if its walkable terrain or not (coz apan ne kuch kuch restriction daale in matrix)
    - Abhi basic cases are passed so we have added neighbours in a temporary "children" array, abhi bhi kuch kuch cases pass karne padenge to go in open list
    - Firstly, if its already in closed list then its processed so no need to add it again
    - Apan fir uss node ki heuristic nikalte (abhi for calculating g, apane jo node closed list mein ie jiske yeh neighbours hein wo inka parent hoga.)
    - Abhi once we have heuristic we check if that node already in open list
        - Agar hua toh we check its heuristic in open list
            - Agar uska current > open list -> toh no change coz we are interested in lower heuristic
            - Agar uska current < open list -> toh fir it means ki uske pehle wale parent ke hissab se iss parent par kum heuristic arahi so we append it

KHATAM

Code:

```
class Node():
    """A node class for A* Pathfinding"""

    def __init__(self, parent=None, position=None):
        #position will be a tupple of (x,y)
        self.parent = parent
        self.position = position
        #cost values
        self.g = 0
        self.h = 0
        self.f = 0

    def __eq__(self, other):
        return self.position == other.position



def astar(maze, start, end):
    #initialise start & end nodes
    start_node = Node(None, start)
    start_node.g = start_node.h = start_node.f = 0
    end_node = Node(None, end)
    end_node.g = end_node.h = end_node.f = 0
```

```
    #initialise closed list & open list
    #closed list - contains processed nodes
    #open list - contains visited but unprocessed nodes
    open_list = []
    closed_list = []

    # Add the start node
    open_list.append(start_node)

    # Loop until you find the end
    while len(open_list) > 0:
        # Get the current node
        current_node = open_list[0]
        current_index = 0
        for index, item in enumerate(open_list):
            if item.f < current_node.f:
                current_node = item
                current_index = index

        #Remove element from open list with lowest cost and add it in
closed list
        open_list.pop(current_index)
        closed_list.append(current_node)

        # Found the goal
        if current_node == end_node:
            path = []
            current = current_node
            while current is not None:
                path.append(current.position)
                current = current.parent
            #as we added path from target to start we need to reverse it
            return path[::-1]
        #If not the target node we generate its children
        children = []
        for new_position in [(0, -1), (0, 1), (-1, 0), (1, 0), (-1, -1),
(-1, 1), (1, -1), (1, 1)]: # Adjacent squares

            # Get node position
            node_position = (current_node.position[0] + new_position[0],
current_node.position[1] + new_position[1])

            # Make sure within range
            if node_position[0] > (len(maze) - 1) or node_position[0] < 0
or node_position[1] > (len(maze[len(maze)-1]) -1) or node_position[1] < 0:
                continue

            # Make sure walkable terrain
            if maze[node_position[0]][node_position[1]] != 0:
                continue
            # Create new node
            new_node = Node(current_node, node_position)
```

```
            # Append
            children.append(new_node)

        # Loop through children
        for child in children:

            # Child is on the closed list
            for closed_child in closed_list:
                if child == closed_child:
                    continue

            # Create the f, g, and h values
            child.g = current_node.g + 1
            child.h = ((child.position[0] - end_node.position[0]) ** 2) +
((child.position[1] - end_node.position[1]) ** 2)
            child.f = child.g + child.h

            # Child is already in the open list
            for open_node in open_list:
                if child == open_node and child.g > open_node.g:
                    continue

            # Add the child to the open list
            open_list.append(child)

def plot (maze,path,start,end):
  for i in path:
    maze[i[0]][i[1]]=2
  maze[start[0]][start[1]] = 3
  maze[end[0]][end[1]] = 4
  for r in maze:
   for c in r:
      if(c==0):
        print('🌱',end = " ")
      elif(c==1):
        print('❌',end = " ")
      elif (c==3):
        print('🐶',end = " ")
      elif(c==4):
        print('🦴',end = " ")
      else:
        print('✅',end = " ")
   print()


def main():

    maze = [[0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
```

```
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

    x1=int(input("Enter x coordinate of start: (< 9)"))
    y1=int(input("Enter y coordinate of start: (< 9)"))
    x2=int(input("Enter x coordinate of end: (< 9)"))
    y2=int(input("Enter x coordinate of end: (< 9)"))

    start = (x1, y1)
    end = (x2, y2)

    path = astar(maze, start, end)
    print(path)
    plot(maze,path,start,end)


if __name__ == '__main__':
    main()
```

Complexity
Time - O(E), where E is the number of edges in the graph
Space - O(V), where V is the total number of vertices.

Alternatively you can use the keyboard shortcut Ctrl+. to trigger the window directly and find an appropriate emoji for what you're trying to say.

#diff between a* and ao*

Assign 3- Selection Sort

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array

```
// C++ program for implementation of selection sort
#include <iostream>
#include <vector>
using namespace std;

void selectionSort(vector<int>&arr,int n)
{
   for(int i=0;i<n-1;i++)
   {
       int min_index = i;
      for (int j=i+1;j<n;j++)
      {
         if(arr[min_index]>arr[j])
         {
            min_index = j;
         }
      }
      swap(arr[i],arr[min_index]);
   }
}
```

```
void display(vector<int>&arr,int n)
{
   for (int i=0;i<n;i++)
   {
      cout<<arr[i]<<" "<<endl;
   }
}

int main()
{
   vector<int>arr;
   int n;
   cout<<"Length of array"<<endl;
   cin>>n;
   for(int i=0;i<n;i++)
   {
      int data;
      cout<<"Enter data"<<endl;
      cin>>data;
      arr.push_back(data);
   }
   selectionSort(arr,n);
   display(arr,n);

}
```

Time Complexity - O(n^2)

Space Complexity - O(n)

#what is greedy algo

Assign 4- N Queens - Backtracking

```
#include <iostream>
using namespace std;
#define N 8

void display(int board[N][N])
{
    for (int i=0;i<N;i++)
    {
        for (int j=0;j<N;j++)
        {
            cout<<board[i][j]<<" ";
        }
        cout<<endl;
    }
}

bool isValid(int board[N][N],int row,int col)
```

```
{
    //no samw row
    for (int i=0;i<col;i++)
    {
        if(board[row][i])
        {
            return false;
        }
    }
    //left upper diagonal
    for (int i=row,j=col;i>=0 && j>=0;i--,j--)
    {
        if(board[i][j])
        {
            return false;
        }
    }
    for (int i=row,j=col;i<N && j>=0;i++,j--)
    {
        if(board[i][j])
        {
            return false;
        }
    }
    return true;
}

bool placeQueens(int board[N][N],int col)
{
    if(col>=N)
    {
        return true;
    }
    else
    {
        for (int i=0;i<N;i++) //row
        {
            if(isValid(board,i,col))
            {
                board[i][col] = 1;
                if(placeQueens(board,col+1))
                {
                    return true;
                }
                board[i][col] = 0;
            }
        }
        return false;
    }
}

void checkQueens()
{
```

```
    //initialise chess board
    int board[N][N];
    for (int i=0;i<N;i++)
    {
        for (int j=0;j<N;j++)
        {
            board[i][j] = 0;
        }
    }
    if(placeQueens(board,0))
    {
        cout<<"yes"<<endl;
        display(board);
    }
    else
    {
        cout<<"Not possible"<<endl;
    }
}

int main()
{
    checkQueens();
}
```

Time Complexity - O(n^3)
Space Complexity - O(n^2)

Assign 4- N Queens - Branch & Bound

https://iq.opengenus.org/8-queens-problem-using-branch-and-bound/

N Queens Problem using Backtracking | Branch and Bound Algorithm Explained

```
/* C++ program to solve N Queen Problem using Branch
and Bound */
#include <iostream>
# include <string.h>
using namespace std;
#define N 8

/* A utility function to print solution */
void printSolution(int board[N][N])
{
      for (int i = 0; i < N; i++)
      {
            for (int j = 0; j < N; j++)
                  cout << " "<< board[i][j];
            cout << "\n";
      }
}
```

```
/* A Optimized function to check if a queen can
be placed on board[row][col] */
bool isSafe(int row, int col, int slashCode[N][N],
                int backslashCode[N][N], bool rowLookup[],
     bool slashCodeLookup[], bool backslashCodeLookup[] )
{
     if (slashCodeLookup[slashCode[row][col]] ||
          backslashCodeLookup[backslashCode[row][col]] ||
          rowLookup[row])
     return false;

     return true;
}

/* A recursive utility function
to solve N Queen problem */
bool solveNQueensUtil(int board[N][N], int col,
     int slashCode[N][N], int backslashCode[N][N],
                                        bool rowLookup[N],
                                   bool slashCodeLookup[],
                                bool backslashCodeLookup[] )
{
     /* base case: If all queens are placed
     then return true */
     if (col >= N)
          return true;

     /* Consider this column and try placing
     this queen in all rows one by one */
     for (int i = 0; i < N; i++)
     {
          /* Check if queen can be placed on
          board[i][col] */
          if ( isSafe(i, col, slashCode,
                         backslashCode, rowLookup,
          slashCodeLookup, backslashCodeLookup) )
          {
               /* Place this queen in board[i][col] */
               board[i][col] = 1;
               rowLookup[i] = true;
               slashCodeLookup[slashCode[i][col]] = true;
               backslashCodeLookup[backslashCode[i][col]] = true;

               /* recur to place rest of the queens */
               if ( solveNQueensUtil(board, col + 1,
                                        slashCode, backslashCode,
               rowLookup, slashCodeLookup, backslashCodeLookup) )
                    return true;
```

```
                /* If placing queen in board[i][col]
                doesn't lead to a solution, then backtrack */

                /* Remove queen from board[i][col] */
                board[i][col] = 0;
                rowLookup[i] = false;
                slashCodeLookup[slashCode[i][col]] = false;
                backslashCodeLookup[backslashCode[i][col]] = false;
           }
      }

      /* If queen can not be place in any row in
           this column col then return false */
      return false;
}

/* This function solves the N Queen problem using
Branch and Bound. It mainly uses solveNQueensUtil() to
solve the problem. It returns false if queens
cannot be placed, otherwise return true and
prints placement of queens in the form of 1s.
Please note that there may be more than one
solutions, this function prints one of the
feasible solutions.*/
bool solveNQueens()
{
      int board[N][N];
      memset(board, 0, sizeof board);

      // helper matrices
      int slashCode[N][N];
      int backslashCode[N][N];

      // arrays to tell us which rows are occupied
      bool rowLookup[N] = {false};

      //keep two arrays to tell us
      // which diagonals are occupied
      bool slashCodeLookup[2*N - 1] = {false};
      bool backslashCodeLookup[2*N - 1] = {false};

      // initialize helper matrices
      for (int r = 0; r < N; r++)
           for (int c = 0; c < N; c++) {
           slashCode[r] = r + c,
                backslashCode[r] = r - c + 7;
           }

      if (solveNQueensUtil(board, 0,
```

```
                                    slashCode, backslashCode,
      rowLookup, slashCodeLookup, backslashCodeLookup) ==
                                                                  false
)
      {
            cout << "Solution does not exist";
            return false;
      }

      // solution found
      printSolution(board);
      return true;
}

// Driver program to test above function
int main()
{
      solveNQueens();

      return 0;
}

// this code is contributed by shivanisinghss2110
```

Assign 5- Chatbot

```
from nltk.chat.util import Chat, reflections
pairs = [
   [
      r'(.*)name?',
      ['Hi myself chatty']
   ],
   [
      r'(.*)age?',
      ['Hi my age is 2 yrs']
   ],
   [
      r"quit",
      [
         "BBye take care. See you soon :) ",
         "It was nice talking to you. See you soon :)",
      ],
   ]
]
def chatty():
```

```
    print('type in lowercase')
    print()
    chat = Chat(pairs,reflections)
    chat.converse()
chatty()
```

OR

```
i = 7
while(i!=0):
    i = int(input("1.Book Plane 2.Book Train 3. Book Taxi\n0 to stop"))
    if(i==1):
        i = int(input("1.Air India 2.Indigo 3.SpiceJet"))
        if(i==1):
            print("Air India website and app is available")
        elif(i==2):
            print("Indigo Website available")
        elif (i==3):
            print("SpiceJet Website available")
        else:
            print("Invalid input, retry")
            break
    elif(i==2):
        i = int(input("1.IRCTC 2.RailYatra 3.BookTrain"))
        if(i==1):
            print("IRCTC website and app is available")
        elif(i==2):
            print("RailYatra Website available")
        elif (i==3):
            print("BookTrain Website available")
        else:
            print("Invalid input, retry")
            break
    elif(i==3):
        i = int(input("1.Ola 2.Uber 3.Meru"))
        if(i==1):
            print("Ola website and app is available")
        elif(i==2):
            print("Uber Website available")
        elif (i==3):
            print("Meru Website available")
        else:
            print("Invalid input, retry")
            break
    elif(i==0):
        print("Thank you")
    else:
        print("Invalid input, retry")
        break
print("Thank you")
```

Ass 6 - Expert Sys

```
Questions=[
    'Do you have cough?',
    'sore throat',
    'problem in breathing',
    'fever',
    'loss of smell',
    'tiredness',
    'chilly',
    'breathlessness',
]
threshold = {
    'mild':30,
    "moderate":50,
    "severe":75
}
def expertSys(Questions,threshold):
    score = 0
    for question in Questions:
        print(question)
        ans = input('y/n?')
        if (ans=='y'):
            level = int(input("On a scale of 0-10 how severe is it?"))
            score+=level
    print()
    if(score>=threshold['severe']):
        print('You are in severe condition')
    elif(score>=threshold['moderate']):
        print('You are in moderate condition')
    elif (score>= threshold['mild']):
        print("Mild symtoms")
    else:
        print("You are fine")
```

Assign 9,10- SalesForce

https://developer.salesforce.com/

Area Calc Ass

Code -
file-> New ->Apex Class -> Name = MethodOverloading

```
public class MethodOverloading {

    public void areaofCircle (Decimal x){
        Decimal Area=3.14*x*x;
        system.debug('Area of circle is'+Area);
    }
```

```
public void areaofRectangle (Decimal x,Decimal y){
        Decimal Area=x*y;
        system.debug('Area of Rectangle is'+Area);
    }
    public void areaofTriangle (Decimal x,Decimal y){
        Decimal Area=0.5*x*y;
        system.debug('Area of Triangle is'+Area);
    }
}
```

In anonymous console

```
MethodOverloading m = new MethodOverloading();
m.areaofCircle(3.2);
m.areaofRectangle(3.2,2);
m.areaofTriangle(5,6);
```

Apex resources

https://www.tutorialspoint.com/apex/apex_overview.htm

https://www.tutorialspoint.com/apex/apex_data_types.htm

https://www.tutorialspoint.com/apex/apex_variables.htm

Salesforce Calculator Ass

file> New->Visualforce Page -> Name = Sample

```
<apex:page controller="Sample">


<apex:form >


        <apex:pageBlock >
        <apex:pageBlockSection >
        <apex:pageBlockSectionItem >
        <apex:outputLabel value="Value 1"/>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:inputText value="{!val1}"/>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:outputLabel value="Value 2"/>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:inputText value="{!val2}"/>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:selectRadio value="{!func}" layout="pageDirection">
                <apex:selectOption itemValue="add" itemLabel="Add"/>
                <apex:selectOption itemValue="sub" itemLabel="Subtract"/>
```

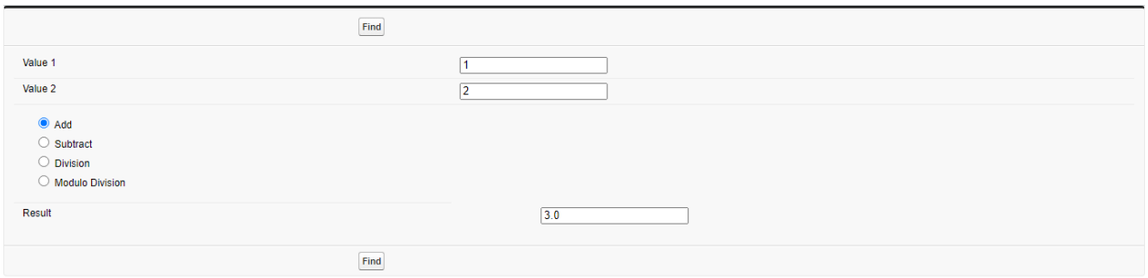
```
                <apex:selectOption itemValue="div" itemLabel="Division"/>
                <apex:selectOption itemValue="mod" itemLabel="Modulo Division"/>
        </apex:selectRadio>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:outputLabel value="Result"/>
        </apex:pageBlockSectionItem>
        <apex:pageBlockSectionItem >
        <apex:inputText value="{!result}" id="res"/><apex:actionStatus id="sts"
startText="Finding..."/>
        </apex:pageBlockSectionItem>
        </apex:pageBlockSection>
        <apex:pageBlockButtons >
        <apex:commandButton value="Find" action="{!find}" reRender="res"  status="sts"/>
        </apex:pageBlockButtons>
        </apex:pageBlock>


</apex:form>


</apex:page>
```

file> New-> Apex Class -> Name = Sample

```
public class Sample
{
        public Double val1 {get;set;}
        public Double val2 {get;set;}
        public Double result {get;set;}
        public String func {get;set;}

        public Sample()
        {
        }

        public void find()
        {
        if(func == 'add')
        {
        result = val1 + val2;
        }
        else if(func == 'sub')
        {
        result = val1 - val2;
        }
        else if(func == 'div')
        {
        result = val1 / val2;
        }
        else
```

```
        {
        Integer temp =  math.mod(Integer.valueOf(val1) , Integer.valueOf(val2));
        result = Double.valueOf(temp);
        }
        }

}
```

Save both files and then click on Preview on Visualforce tab



Calculator with validation

```
public class Sample
{
   public Double val1 {get;set;}
   public Double val2 {get;set;}
   public String result {get;set;}
   public String func {get;set;}

   public Sample()
   {
   }

   public void find()
   {
      if(func == 'add')
      {
         result = String.valueOf(val1 + val2);
      }
      else if(func == 'sub')
      {
          result = String.valueOf(val1 - val2);
      }
      else if(func == 'div')
      {
         if(val2==0){
            result='Don\'t divide by zero';
         }else{
            result = String.valueOf(val1 / val2);
         }
      }
      else
      {
         if(val2==0){
            result='Don\'t divide by zero';
         }else{
          Integer temp =  math.mod(Integer.valueOf(val1) , Integer.valueOf(val2));
```

```
            result = String.valueOf(temp);
          }
       }
    }

}
```

Calc with scientific

APXC

```
public class calculatorDemo
{
    public Double val1 {get;set;}
    public Double val2 {get;set;}
    public Double result {get;set;}
    public String func {get;set;}

    public calculatorDemo()
    {
        this.val1 = 1;
        this.val2 = 1;
        this.result = 0;
    }

    public void find()
    {
        if(func == 'add')
        {
            result = val1 + val2;
        }
        else if(func == 'sub')
        {
             result = val1 - val2;
        }
        else if(func == 'div')
        {
             result = val1/val2;
        }
        else if (func == 'sin'){
            result = math.sin(val1*0.0175);
        }
        else if( func == 'cos'){
            result = math.cos(val1*0.0175);
        }
        else if(func == 'tan'){
            result = math.tan(val1*0.0175);
        }
        else if(func == 'log'){
            result = math.log10(val1);
        }
        else
        {
             Integer temp =  math.mod(Integer.valueOf(val1) , Integer.valueOf(val2));
             result = Double.valueOf(temp);
        }
    }
```

```
}
```

vfp

```
<apex:page controller="calculatorDemo">

<apex:form >

   <apex:pageBlock >
      <apex:pageBlockSection >
         <apex:pageBlockSectionItem >
            <apex:outputLabel value="Value 1"/>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:inputText value="{!val1}"/>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:outputLabel value="Value 2"/>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:inputText value="{!val2}"/>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:selectRadio value="{!func}" layout="pageDirection">
               <apex:selectOption itemValue="add" itemLabel="Add"/>
               <apex:selectOption itemValue="sub" itemLabel="Subtract"/>
               <apex:selectOption itemValue="div" itemLabel="Division"/>
               <apex:selectOption itemValue="mod" itemLabel="Modulo Division"/>
               <apex:selectOption itemValue="sin" itemLabel="Sin"/>
               <apex:selectOption itemValue="cos" itemLabel="Cos"/>
               <apex:selectOption itemValue="tan" itemLabel="Tan"/>
               <apex:selectOption itemValue="log" itemLabel="Logarithm"/>
            </apex:selectRadio>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:outputLabel value="Result"/>
         </apex:pageBlockSectionItem>
         <apex:pageBlockSectionItem >
            <apex:inputText value="{!result}" id="res"/><apex:actionStatus id="sts"
startText="Finding..."/>
         </apex:pageBlockSectionItem>
      </apex:pageBlockSection>
      <apex:pageBlockButtons >
         <apex:commandButton value="Find" action="{!find}" reRender="res"
status="sts"/>
      </apex:pageBlockButtons>
   </apex:pageBlock>

</apex:form>

</apex:page>
```

Student sys

```
public class Stud {
   public static Integer cnt = 0;
   public static List<Stud> students = new List<Stud>{};
   public String name = '';

   public Stud(String name1){
      this.name = name1;
      cnt++;
  students.add(this);
   }

   static public void display(){
      for(Stud i:students){
         System.debug(i.name);
      }
   }

   static public void upd(String n1, String n2){
      for(Integer i = 0; i < cnt; i++){
         if(students[i].name == n1){
            students[i].name = n2;
         }
      }
   }

   static public void del(String name){
      for(Integer i = 0; i < cnt; i++){
         if(students[i].name == name){
            students.remove(i);
         }
      }
      cnt--;
   }
}
//public static void ...
```

KVM

https://www.tecmint.com/install-kvm-on-ubuntu/

sudo virt-manager : To open the virtual machine manager window (additional command not there in article)

https://phoenixnap.com/kb/ubuntu-install-kvm