Frequency of a Alphabets in a Word

1. Start
2. str=User input Word
3. str converted toLowerCase()
4. Initialise ctr array=array of 26 int
5. for loop from i=0 to str.length-1
    a. k=(int)(character at position i of str)
    b. ctr[k] incremented
6. Print frequency of alphabets
    a. for loop from i=0 to ctr.length-1
    b. if ctr[i]!=0
    c. then (char)(97+i)
    d. Print ctr[i]

Removal of Consecutive Occurrences in a Word

1. Start
2. Scanner object initialised
3. str=User input Word
4. String output initialised
5. for loop from i=0 to str.length-1
    a. if str.substring(i+1) has at 0^{th} position str.charAt(i)
    b. then continue loop
    c. else output=output+str.charAt(i)
6. Print output

Decryption of an Encrypted String

1. Start
2. shift=User input shift value
3. str=User input encrypted string
4. Print decrypt(shift,str)
    a. str passed as parameter code, function String decrypt(int shift,String code)
    b. ptext string initialised
    c. code converted toUpperCase(), trim(), removed all spaces
    d. for loop from i=0 to code.length-1
        i. char ch initialised to code.charAt(i)
        ii. n=(int)ch
        iii. n=n-64
        iv. ptext=ptext+decrypt(ch,shift)
            1. ch passed as parameter c, function char decrypt(char c,int shift)
            2. alpha=char array of 27 elements
            3. beta=char array of 27 elements
            4. k=shift
            5. char c2 initialised
            6. for loop from i=0 to 26
                a. alpha[i+1]=(char)(65+i)
            7. h=1
            8. for loop from k to beta.length-1
                a. beta[k++]=alpha[h]
            9. for loop from k=1 to shift-1, h increments with loop
                a. beta[k++]=alpha[h]
            10. for loop from i=1 to beta.length-1
                a. if beta[i]==c
                b. then c2=alpha[i]
            11. c2 returned
        v. ptext now contains decrypted string of char
    e. replace every instance of “QQ” with “ “ in ptext
    f. ptext returned i.e. printed

Deleting a Word from a Sentence and extra Spaces

1. Start
2. Scanner object initialised
3. str=User input sentence
4. char punc initialised
5.
    a. if str has '.' has any position>0
    b. then
        i. str= substring of str until '.'
        ii. punc='.'
    c. else if str has '?' has any position>0
    d. then
        i. str= substring of str until '?'
        ii. punc='?'
    e. else str has '!' has any position>0
    f. then
        i. str= substring of str until '!'
        ii. punc='!'
6. Scanner object closed
7. Scanner object reinitialised
8. word=User input word to be deleted
9. Scanner object closed
10. Scanner object reinitialised with str string
11. output string initialised
12. ctr=0,k=0
13. Scanner object set delimiter to space
14. while scanner object has tokens
    a. part=scanner object's next token/word from sentence
    b. if part equals ""
    c. then continue
    d. ctr incremented
    e. if par equalsIgnoreCase with word is false
    f. then output=output+part+" "
    g. else
    h. then k=ctr
15. output trim()
16. Print output+punc
17. Print k/position of word deleted

Fibonacci String

1. Start
2. FiboString object initialised
    a. string x,y,z initiliased
    b. n stores limit
    c. x=”a”,y=”b”,z=x+y
3. FiboString Object’s accept() function invoked
    a. n=User input limit of Fibonacci string
4. FiboString Object’s generate() function invoked
    a. s0=x,s1=y,s2=z
    b. Print s0+”,”+s1
    c. for loop from i=2 to limit
        i. s2=s1+s0
        ii. Print “,”+s2
        iii. s0=s1,s1=s2

Upper Triangular Matrix Test

1. Start
2. Scanner object initialised
3. n=User input order of square matrix
4. Initialised dda=double dimensional array of n x n
5. for loop from i=0 to n-1
    a. for loop j=0 to n-1
        i. dda[i][j]=User input number
6. flag=true
7. for loop from i=0 to n-1
    a. for loop from j=0 to i
        i. if dda[i][j]!=0
        ii. then flag=false
8.
    a. if flag
    b. then Print Upper Triangular Matrix
    c. else
    d. then Print Not Upper Triangular Matrix

Saddle Point of Matrix

1. Start
2. Scanner Object initialised
3. n=User input order of square matrix
4. dda=double dimensional array of n x n int
5. for loop from i=0 to n-1
    a. for loop from j=0 to n-1
        i. dda[i][j]=User input number
6. ref[]=saddle point index of dda
    a. function static int[] indexOfSaddlePoint(int[][] dda)
    b. ref[]=null
    c. for loop from i=0 to dda[0].length-1
        i. x=indexOfMin(dda[i])
            1. function static int indexOfMin(int[] ar)
            2. min=ar[0]
            3. k=0
            4. for loop from i=0 to ar.length-1
                a. if ar[i]<min then min=ar[i],k=i
            5. k returned
        ii. ar=array of dda[0].length
        iii. for loop from j=0 to ar.length-1
            1. ar[j]=dda[j][x]
        iv. y=indexOfMax(ar)
            1. function static int indexOfMax(int[] ar)
            2. max=0
            3. k=0
            4. for loop from i=0 to ar.length-1
                a. if ar[i]>max then max=ar[i],k=i
            5. k returned
        v. if dda[x][y]==dda[i][y] then ref[0]=y,ref[1]=x,break
    d. ref returned
7.
    a. if ref!=null
    b. then Print ref[0]+ref[1]+dda[ref[0]][ref[1]]
    c. else
    d. then Print No Saddle Point

Magic Square Matrix Test

1. Start
2. Scanner object initialised
3. n=User input order of square matrix
4. Initialised dda=double dimensional array of n x n
5. for loop from i=0 to n-1
    a. for loop j=0 to n-1
        i. dda[i][j]=User input number
6. flag=true
7. sum1=0
8. for loop from i=0 to n-1
    a. sum1=sum1+dda[0][i]
9. for loop from i=0 to n-1, until flag==true
    a. sum2=0
    b. for loop from j=0 to n-1
        i. sum2=sum2+dda[i][j]
    c. if sum2!=sum1
        i. flag=false
        ii. break loop
10. s=0
11. for loop from i=0 to n-1
    a. s=s+dda[i][i]
12. if s!=sum1 then flag=false
13.
    a. if flag
    b. then Print Magic Square Matrix
    c. else
    d. then Print Not Magic Square Matrix

Boundary of Matrix

1. Start
2. Scanner object initialised
3. m=User input number of rows
4. n=User input of number of columns
5. Initialised dda=double dimensional array of m x n
6. for loop from i=0 to m-1
    a. for loop j=0 to n-1
        i. dda[i][j]=User input number
7. for loop from i=0 to n-1
    a. Print "\n"
    b. for loop j=0 to n-1
        i. if i==0||j==0||i==(m-1)||j==(n-1)
        ii. then Print dda[i][j]+"|"
        iii. else
        iv. then Print "|"

Diagonals of Matrix

1. Start
2. Scanner object initialised
3. M=User input order of Square Matrix
4. if M>2||M>10 then terminate/return
5. Initialised dda=double dimensional array of M x M
6. for loop from i=0 to M-1
    a. for loop j=0 to M-1
        i. dda[i][j]=User input number
7. flag=true
8. for loop from i=0 to M-1
    a. for loop j=0 to M-1
        i. if dda[i][j]!=dda[j][i]
        ii. then flag=false, break loop
9. Print original Matrix / for loop from i=0 to M-1
    a. for loop j=0 to M-1
        i. Print dda[i][j]+"|"
10. if flag then Print SYMMETRIC else then Print NOT SYMMETRIC
11. sum1=0,sum2=0
12. Print left diagonal / for loop from i=0 to M-1
    a. for loop j=0 to i
        i. Print "\t"
    b. Print dda[i][i]+"\n"
    c. sum1=sum1+dda[i][i]
13. Print sum1 / Sum of left diagonal
14. Print right diagonal / for loop from i=M-1 to 0
    a. for loop j=i to 0
        i. Print "\t"
    b. Print dda[M-1-i][i]+"\n"
    c. sum2=sum2+dda[M-1-i][i]
15. Print sum2 / Sum of Right Diagonal

Class Matrix

1. Start
2. Scanner object initialised
3. mm=User input number of rows
4. nn=User input number of columns
5. Matrix Object a,b,c initialised (default constructors invoked)
6. Matrix a fillarray() invoked
    a. for loop from i=0 to m-1 (mm-1 in this case)
        i. for loop from i=0 to n-1 (nn-1 in this case)
            1. ar[i][j]=User input number
7. Matrix b fillarray() invoked
    a. Same as 6.a
8. Matrix c=a.subMat(b)
    a. b passed as argument, function Matrix subMat(Matrix A)
    b. new Matrix initialised of dimensions A.m,A.n (b.m,b.n in this case)
    c. if this.m==A.m && this.n==A.n
    d. then
        i. for loop from i=0 to A.m-1
            1. for loop from j=0 to A.n-1
                a. b.ar[i][j]=this.ar[i][j]-A.ar[i][j]
    e. b returned
    f. C is subtracted matrix of B from A
9. for loop from i=0 to c.m-1
    a. for loop from j=0 to c.n-1
        i. Print ar[i][j]+"|"
    b. Print "\n"

Class Time

1. Start
2. Scanner Object initialised
3. h1=User input hour
4. m1=User input minute
5. Time t1 initialised
    a. hrs=0,min=0
6. Time t1 getTime() invoked with h1,m1
    a. function void getTime(int h,int m)
    b. hrs=h,min=m (h1,m1 in this case)
7. h2=User input hour
8. m2=User input minute
9. Time t2 initialised
    a. Same as 5.a
10. Time t2 getTime() invoked with h2,m2
    a. Same as 6.a,6.b
11. Time t3 initialised
12. t3 sumTime() invoked with t1,t2
    a. function Time sumTime(Time t1,Time t2)
    b. Time t3 initialised
    c. t3.hrs=t1.hrs+t2.hrs
    d. t3.hrs+=(t1.min+t2.min)/60
    e. t3.min=(t1.min+t2.min)%60
    f. t3 returned
13. t3 printTime() invoked
    a. function void printTime()
    b. Print h+":"+m (t3.h,t3.m in this case)

Class Array

1. Start
2. Array Object A,B initialised
    a. ar array initialised of length 100
    b. n=0
    c. num=0
3. A object's getArray() invoked
    a. function void getArray()
    b. for loop from i=0 to n-1
        i. this.ar[i]=User input number
4. k=User input number
5. Print k+" "+a.process(a,k)
    a. a passed as argument to B, function int process(Array B,int k)
    b. this.num=k,ctr=0
    c. for loop from i=0 to B.n-1 (a.n-1 in this case)
        i. if B.ar[i]==this.num
        ii. then ctr++
    d. ctr returned and printed
6. B object's getArray invoked
    a. Same as 3.a,3.b
7. Array Object C initialised
8. c=a object's merge() invoked with argument b
    a. function Array merge(Array a)
    b. b.n=this.n+a.n,k=0
    c. for loop from i=0 to this.n-1
        i. b.ar[k++]=this.ar[i]
    d. for loop from i=0 to a.n-1
        i. b.ar[k++]=a.ar[i]
    e. b returned
9. c Object's display() invoked
    a. for loop from i=0 to this.n-1
        i. if i%4!=0
        ii. then Print ar[i]+"\t"
        iii. else
        iv. then Print "\n"+ar[i]+"\t"

Class Transarray

1. Start
2. m=User input array row length
3. n=User input array column length
4. Transarray Object A,C initialised (parameterised constructors invoked)
    a. Transarray object whose array have dimensions m x n
    b. this.arr=double dimensional array of M x N int
5. A object's fillarray() invoked
    a. function void fillarray()
    b. Scanner object initialised
    c. for loop from i=0 to this.M-1
        i. for loop from j=0 to N-1
            1. this.arr[i][j]=User input number
6. C object's transpose() invoked with Transarray A as argument
    a. function void transpose(Transarray A)
    b. New Transarray B object initialised
    c. for loop from i=0 to B.M-1
        i. for loop from j=0 to B.N-1
            1. B.arr[j][i]=A.arr[i][j]
    d. this.arr=B.arr,this.M=B.M,this.N=B.N
7. C object's disparray() invoked
    a. function void disparray()
    b. for loop from i=0 to this.M-1
        i. for loop from j=0 this.N-1
            1. Print this.arr[i][j]

Class Collection

1. Start
2. Scanner object initialised
3. n1=User input length of Collection 1
4. Collection Object a initialised (parameterised constructor invoked)
    a. ar=array of 100 int
    b. len=argument
5. a object's input() invoked
    a. function void input()
    b. Scanner object initialised
    c. for loop from i=0 to len-1
        i. ar[i]=User input number
6. n2=User input length of Collection 2
7. Collection Object b initialised
    a. Same as step 4
8. b object's input invoked
    a. Same as step 5
9. Collection Object c initialised
    a. Same as step 4
10. c=a object's common(), b passed as argument
    a. function Collection common(Collection c)
    b. ctr=0
    c. for loop from i=0 to this.len-1
        i. for loop from j=0 to c.len (b.len in this case)
            1. if this.ar[i]==c.ar[j]
            2. then ctr++
    d. Collection object A initialised
    e. k=0
    f. for loop from i=0 to this.len-1
        i. for loop from j=0 to c.len (b.len in this case)
            1. if this.ar[i]==c.ar[j]
            2. A.ar[k++]=this.ar[i]
    g. Collection A returned
11. c Object's display() invoked
    a. Scanner Object initialised
    b. for loop from i=0 to len-1
        i. Print ar[i]