KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

		L	T	P	C
17MMP211	**OPTIMIZATION TECHNIQUES - PRACTICAL**	**0**	**0**	**4**	**2**

List of Practical:

1. Solution for a system of equations- Simplex method.
2. Decision Making with minimax criteria.
3. Decision Making under risk.
4. Travelling salesman problem to find the shortest path.
5. Write a C program to calculate the minimum cost using North West Corner Rule.
6. To calculate the EOQ for purchasing model without shortage using C program.
7. To calculate the EOQ for manufacturing model without shortage using C program.
8. To calculate the EOQ for manufacturing model with shortage using C program.
9. To calculate the EOQ for purchasing model with shortage using C program.
10. Probabilistic Model-EOQ.

Content

KAHE

EX.NO:1	FAMILY OF DIFFERENTIAL EQUATIONS

AIM:

To write a c program to find the solution for LPP using simplex method.

ALGORITHM:

STEP 1: Start the program.

STEP 2: Declare the variable required for the program.

STEP 3: Print the maximum and minimum choice.

STEP 4: Get the pivotal row and column and pivotal element to find the solution.

STEP 5: Find the new equation for s1 and s2.

STEP 6: The requesting is printout.

STEP 7: Display the result.

STEP 8: Stop the process.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
float a[10][10]={0},b[10],d[10],x[10][3]={3};
int m,n,s=1;
void main()
{
int i,j,m1,n1,c[10]={0};
```

```
float m2,d[10],s=0;
void table();
clrscr();
printf("\n \t \t ***SIMPLEX METHOD***\n ");
printf("\n 1.maximum \n 2.minimum \n choice");
scanf("%d",&m1);
printf("enter the coefficient in the main equation:");
scanf("%d",&n);
printf("enter the coefficient:");
for(i=1;i<=n;i++)
{
scanf("%d",&c[i]);
if(m1==2)
c[i]=-1*c[i];
}
printf("\n enter the number of constraints:");
scanf("%d",&m);
printf("\n enter the coefficient one by one:");
for(i=1;i<=m;i++)
{
printf("enter the coeffient of the constraints %d:",i);
for(j=1;j<=n;j++)
scanf("%f",&a[i][j]);
printf("enter the contant:");
```

```
scanf("%f",&a[i][0]);
if((d[i]!=0)&&(d[j]>0))
if(d[j]<m2)
{
m2=d[j];
n1=j;
}
}
m2=a[n1][i];
printf("\n pivotal column:y %d",i);
printf("\n pivotal row: %d",n1);
printf("\n pivotal element: %3.2f",m2);
getch();
for(j=0;j<=m+n;j++)
a[n1][j]=a[n1][j]/m2;
m2=i;
x[n1][0]=c[m2];
x[n1][1]=m2;
for(i=1;i<m;i++)
if(n1!=i)
{
s=a[i][m2];
for(j=0;j<=m+n;j++)
a[i][j]=a[i][j]-(s*a[i][j]);
```

KAHE

```
}
goto line;
{
if(m1==m2)
b[0]=-1*b[0];
for(i=1;i<=m;i++)
x[i][2]=a[i][0];
printf("\n when");
for(i=1;i<=m;i++)
{
if(x[i][j]<=n)
printf("\n \t x%1.0f = 3.3f",x[j][i],x[1][2]);
}
printf("\n \n \n z=\t %5.3f",b[0]);
line;
getch();
}
int s1,s2;
clrscr();
printf("\n table %d \n ",s);
s++;
line();
printf("\n <b \t \t x");
for(s1=1;s1<=m;s1++)
```

```
{
printf("\n \n %1.0f \t "x[s1][0],x[s1][1]);
for(s2=0;s2<=m+n;s2++)
printf("%2.1f",a[s1][s2]);
}
printf("\n");
line();
printf("\n \t z \t ");
for(s1=0;s1<=m+n;s1++)
printf("%2.1f",b[s1]);
}
void line();
int s1;
for(s1=1;s1<=(m+n+3)*7;s1++)
printf("*");
}
```

OUTPUT:

KAHE

EX.NO:2	DECISION MAKING WITH MINIMAX CRITERIA

QUESTION:

Find decision making under risk

AIM:

To Write a program to find decision making under risk.

ALGORITHM:

STEP 1: Start the process.

STEP 2: Include the necessary header file.

STEP 3: Declare the variable in intdatatype.

STEP 4: Print the row,column and matrix.

STEP5: Using the for loop statement,

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

for(j=1;j<=m;j++)

STEP 6: Print the greatest value first and second rows.

STEP 7: Find the minimax value.

STEP 8: Stop the process.

KAHE

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int i,j,m,n,a[10][10],p,q;
clrscr();
printf("\n enter the number of rows and columns:");
scanf("%d %d",&m, &n);
printf("\n enter the matrix:");
{
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
scanf("%d",&a[i][j]);
}
printf("\n enter the greatest value of 1st row p:");
scanf("%d",^&p);
printf("\n enter the greatest value of 2nd row q:");
scanf("%d",&q);
if(p<q)
{
printf("\n the minimax value is:%d",p);
}
```

```
else
{
printf("the minimax value is:%d",q);
}
getch();
}
```

OUTPUT

KAHE

EX.NO:3	DECISION MAKING UNDER RISK

QUESTION:

Find decision making under risk

AIM:

Write a C program to find decision making under risk.

ALGORITHM:

STEP 1: Start the process.

STEP 2: Declare the necessary header file.

STEP 3: Declare the variable.

STEP 4: Calculate the Ua,Pa,Pb,Ub,P1,P2 using this formula

P1=Ua*Pa;

P2=Ub*Pb;

STEP 5: Display the result.

STEP 6: Stop the process.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
```

```
{
float d,ed,c1,c3,q;
clrscr();
printf("\n ***INVENTORY CONTROL***\n");
printf("\n enter the setup cost c3=");
scanf("%f",&c3);
printf("\n enter the demand d=");
scanf("%f"&d);
printf("\n enter the carrying cost c1=");
scanf("%f",&c1);
printf("\n purchasing problem without shortage");
ed=(2*d*c3);
q=sqrt(ed/c1);
printf("\n the economic quantity= %f",q);
getch();
}
```

KAHE

EX.NO:4	ASSIGNMENT PROBLEM

QUESTION:

Find maximum cost using assignment problem.

AIM:

To write a C program to find maximum cost using assignment problem.

ALGORITHM:

STEP 1: Start the process.

STEP 2: Declare the variables required for the program.

STEP 3: Get the value of last matrix and assign the matrix do other temporary matrix to noted.

STEP 4: Find the minimum value each row and column find the row minimum matrix displays it

STEP 5: Do the allocation in the result and matrix corresponding allocation in the cost matrix is added.

STEP 5: Stop the process.

KAHE

PROGRAM:

```
clear;
L=1.0;
W=4.0;
T=10.;
k=200;
dt=T/k;
n=10.;
dx=L/n;
m=20.;
dy=W/m;
velx=.1;
vely=.4;
decay=.0;
for i=1:n+1
x(i)=(i-1)*dx;
for j=1:m+1
y(j)=(j-1)*dy;
u(i,j,1)=0.;
end
end
for k=1:k+1
time(k)=(k-1)*dt;
for j=1:m+1
```

KAHE

```
u(1,j,k)=.0;
end
for i=1:n+1
u(i,1,k)=(i<=(n/2+1))*(k<26)*5.0*sin(pi*x(i)*2)+(i>(n/2+1))*.1;
end
end
for k=1:k
for i=2:n+1;
for j=2:m+1;
u(i,j,k+1)=(1-velx*dt/dx-vely*dt/dy-decay*dt)*u(i,j,k)+velx*dt/dx*u(i-1,j,k)+vely*dt/dy*u(i,j-1,k);
end
end
end
mesh(x,y,u(:,:,k)')
```

KAHE

EX.NO:5	NORTH WEST CORNER RULE

QUESTION:

Use c program to find the solution of north west corner rule.

AIM:

To write a c program using north west corner rule.

ALGORITHM:

STEP 1: Start the process.

STEP 2: Declare the variable. Get the number of rows and columns of the matrix .

STEP 3: Using loop increment I and j values oneby one and get the matrix a[i][]j.

STEP 4: Using loop I values one by one and get the availability ofthe matrix ava[i].

STEP 5: Initialize n1,m1,m2 equal to zero.

STEP 6: Check whether req[j]>ava[i];

STEP 7: Initialize tp[m2][i]=ava[i];

Tp[m2][0]=a[i][j];

Calculate req[j]=req[i]-ava[i];

elseava[i]=ava[j]-req[j];

STEP 8: Terminate the program.

PROGRAM:

```
function V=elongation(t)
%Function elongation has input variable
    t and output variable V
%It gives the bacterium volume after
   time t: V=0.4+0.02*t
V=0.4+0.02*t;
elongation(4)
```

KAHE

EX.NO:6	PURCHASING PROBLEM WITHOUT SHORTAGE

QUESTION:

Compute the purchasing problem without shortage.

AIM:

To write a C program to find purchasing problem without shortage.

ALGORITHIM:

STEP1: Start the process.

STEP2: Include necessary header file.

STEP3: Declare a float value q,d,c1,ed,c3.

STEP4: Find the value of ed,is calculate ed=(2*d*c3).

STEP5: Find the sqrt(ed/c1).

STEP6: Print the result.

STEP7: Stop the process.

KAHE

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float d,c1,eq,c2,,k,q1,c3,q;
clrscr();
printf("\n ***INVENTORY CONTROL***\n");
printf("\n enter the setup cost c3=");
scanf("%f",&c3);
printf("\n enter the demand d=");
scanf("%f"&d);
printf("\n enter the carrying cost c1=");
scanf("%f",&c1);
printf("\n enter the production rate k=");
scanf("%f",&k);
printf("\n manufacturing problem without shortage");
q=(2*d*c3)/c1;
q1=(k/(k-d));
eq=sqrt(q*q1);
printf("\n the economic quantity= %f",eq);
getch();
}
```

EX.NO:7	MANUFACTURING PROBLEM WITH SHORTAGE

QUESTION:

Calculate EOQ manufacturing problem without shortage using C program

AIM:

To calculate EOQ manufacturing problem without shortage using C program..

ALGORITHIM:

STEP1: Start the process.
STEP2: Include necessary header file.
STEP3: Declare the variables.
STEP4:.Calculate q1,q,eq using the formula

q =(2*d*c3)/c1;
q1 =(k-/(c-d));

eq=srqt(q*q1);
STEP6: Print the eoq.
STEP7: Display the result.
STEP7: Stop the process.

KAHE

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float d,c1,eq,c2,,k,q1,c3,q,q2;
clrscr();
printf("\n ***INVENTORY CONTROL***\n");
printf("\n enter the setup cost c3=");
scanf("%f",&c3);
printf("\n enter the demand d=");
scanf("%f"&d);
printf("\n enter the carrying cost c1=");
scanf("%f",&c1);
printf("\n enter the production rate k=");
scanf("%f",&k);
printf("\n enter the shortage cost c2=");
scanf("%f",&c2);
printf("\n manufacturing problem with shortage");
q=(2*d*c3)/c1;
q1=(c1+c2)/c2;
```

```
q2=(k/(k-d));

eq=sqrt(q*q1*q2);

printf("\n the economic quantity= %f",eq);

getch();

}
```

OUTPUT:

KAHE

EX.NO:8	PURCHASING PROBLEM WITH SHORTAGE

QUESTION:

Calculate EOQ for purchasing model with shortage using C program

AIM:

To calculate EOQ for purchasing model with shortage using C program.

ALGORITHIM:

STEP1: Start the process.

STEP2: Include necessary header file.

STEP3: Declare a float value q,d,c1,c2,q1,c3 in type of float.

STEP4: Print one setup cost,demand,carryingcost,shortage cost.

STEP5: Calculate q1,q and eoq using the formula

q =(2*d*c3)/c1.

q1 =(c1*c2)/c2.

STEP6: Print the eoq and display the result.

STEP7: Stop the process.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float d,c1,eoq,c2,q1,c3,q;
clrscr();
printf("\n ***INVENTORY CONTROL***\n");
printf("\n enter the setup cost c3=");
scanf("%f",&c3);
printf("\n enter the demand d=");
scanf("%f"&d);
printf("\n enter the carrying cost c1=");
scanf("%f",&c1);
printf("\n enter the shortage cost c2=");
scanf("%f",&c2);
printf("\n purchasing problem with shortage");
q=(2*d*c3)/c1;
q1=(c1+c2)/c2;
eoq=sqrt(q*q1);
printf("\n the economic quantity= %f",eoq);
getch();
}
```

KAHE

EX.NO:9	PROBABILISTIC MODEL

QUESTION:

Find a probabilistic model using C program .

AIM:

To write a C program to find a probabilistic model.

ALGORITHIM:

STEP1: Start the process.

STEP2: Include necessary header file.

STEP3: Declare the variables.

STEP4:.Calculate c1,c2,c3,p using the formula

c2=c3-c1;

p=c2/(c1+c2);

STEP5: Display the result.

STEP6: Stop the process.

```
#include<conio.h>
#include<math.h>
void main()
{
float c1,c2,c3,p;
clrscr();
printf("\n enter the holding cost c1=");
scanf("%f",&c1);
printf("\n enter the selling cost c3=");
scanf("%f",&c3);
c2=c3-c1;
printf("\n enter the carrying cost:%f",c2);
p=c2/(c1+c2);
printf("\n enter the probabilistic eoq is %f",p);
getch();
}
```

KAHE

OUTPUT :

KAHE

EX.NO:9	DECISION MAKING UNDER RISK

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float ua,ub,pa,pb,p1,p2;
clrscr();
printf("\n enter the utitlity value ua:");
scanf("%f",&ua);
printf("\n enter the probability value pa:");
scanf("%f",&pa);
p1=ua*pa;
printf("\n the expected utitlity is:%f",p1);
printf("\n enter the utitlity value is:%f:",p1);
scanf("%f",&ub);
printf("\n enter the probability value pb:");
scanf("%f",&pb);
printf("\n the expected utitlity is:%f",p2);
if(p1>p2)
{
```

```
printf("\n p1 is the best choice invested:");
}
Else
{
printf("\n p2 is the best choice invested:");
}
getch();
}
```

OUTPUT :

KAHE