*Lab 1.1 (2 Points)*

*Ans: SumAll.m*

function y = SumAll( varargin )

if nargin == 0

y = 0;

else

temp = cell2mat(varargin);

y = sum(temp(:));

end

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Lab 1.2 (2 Points)*

*Ans: rsearch.m + rnd\_plot.m + rnd\_search.m + two\_peaks.m*

代码附在最后

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Lab 1.3 (6 Points)*

*Exercise 1: File Operations –Simple Database (3 Points)*

*Ans: people.m*

function people(file,varargin)

people = [];

switch varargin{1}

case 'reset'

people = [];

save(file,'people');

case 'list'

load(file);

for i = 1:length(people)

disp([people(i).name, ' ', num2str(people(i).age)])

end

case 'insert'

for i = 1:(length(varargin) - 1)/2

temp.name = varargin{2\*i};

temp.age = varargin{2\*i + 1};

people = [people, temp];

save(file,'people');

end

case 'remove'

load(file);

index = [];

for i = 1:length(people)

for j = 1:(length(varargin) - 1)/2

if strcmpi(people(i).name,varargin{2\*j}) & people(i).age == varargin{2\*j + 1}

index = [index, i];

end

end

end

people(index) = [];

save(file,'people');

otherwise

fprintf('error');

end

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Exercise 2: Rotating Polygon (3 Points)*

*Ans: rotating\_ploygon.m*

clear;clc;

X = [-2 -5 0 5 2];

Y = [-5 3 3 3 -2];

h = fill(X,Y,'w');

thetad = 1;

R = [cosd(thetad) -sind(thetad); sind(thetad) cosd(thetad)];

C = repmat([0 0], length(X), 1)';

axis([-10 10 -10 10])

for k=1:360

V = get(h,'Vertices')'; % get the current set of vertices

V = R\*(V - C) + C; % do the rotation relative to the centre of the square

set(h,'Vertices',V'); % update the vertices

pause(0.01);

end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Exercise 3: Path (2 Bonus Points)*

*Ans: findpath.m + findpath\_go.m*

\*\*\*\*\*\*\*\*findpath.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear;clc

figure

axis([0 20 0 20])

dots.x = round(rand(1, 100)\*20);

dots.y = round(rand(1, 100)\*20);

plot(dots.x, dots.y, 'ro','MarkerFaceColor','r');

grid on

hold on

temp = 0;

h = [];

i = 0;

while temp~= 20

i = i + 1;

set(h,'visible','off');

[dot, h] = findpath\_go(dots);

temp = dot.y;

end

disp(['尝试', num2str(i), '次到达顶部']);

\*\*\*\*\*\*\*\*findpath\_go.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [dot, h] = findpath\_go(dots)

dot.x = round(rand(1, 1)\*20);

dot.y = 0;

h = plot(dot.x, dot.y, 'bo','MarkerFaceColor','b');

while dot.y ~= 20

index = find(dots.x == dot.x);

if ismember(dot.y + 1, dots.y(index)) == 1

indexl = find(dots.y == dot.y);

if ismember(dot.x - 1, dots.x(indexl)) == 1 & ismember(dot.x + 1, dots.x(indexl)) == 1

return

elseif ismember(dot.x - 1, dots.x(indexl)) == 0 & dot.x - 1 >= 0

dots.x = [dots.x, dot.x];

dots.y = [dots.y, dot.y];

dot.x = dot.x - 1;

elseif ismember(dot.x + 1, dots.x(indexl)) == 0 & dot.x + 1 <= 20

dots.x = [dots.x, dot.x];

dots.y = [dots.y, dot.y];

dot.x = dot.x + 1;

else

break;

end

else

dots.x = [dots.x, dot.x];

dots.y = [dots.y, dot.y];

dot.y = dot.y + 1;

end

h = [h, plot(dot.x, dot.y, 'bo','MarkerFaceColor','b')];

end

end

*rsearch.m + rnd\_plot.m + rnd\_search.m + two\_peaks.m*

\*\*\*\*\*\*\*\*rsearch.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear all

x=0:0.02:0.8;

y=x;

[CData.xi,CData.yi]=meshgrid(x,y);

for j=1:length(x)

for k=1:length(y)

CData.zi(j,k)=two\_peaks([x(j),y(k)]);

end

end

x0=[0.1 0.1]';

B.lb=[0 0]';

B.ub=[0.8 0.8]';

[x\_opt,f\_opt]=rnd\_search('two\_peaks',x0,B,CData,80,1e-5);

fprintf('Final solution: %f %f\n',x\_opt(1),x\_opt(2));

disp(['Function value: ',num2str(f\_opt)]);

\*\*\*\*\*\*\*\*rnd\_plot.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function rnd\_plot(CData,X,x,mode,varargin)

contour(CData.xi,CData.yi,CData.zi,20);

colormap([0 0 1]);

hold on;

plot(X{1}(1),X{1}(2),'co','MarkerSize',5);

for j=2:length(X)

plot(X{j}(1),X{j}(2),'co','MarkerSize',5);

plot([X{j}(1) X{j-1}(1)],[X{j}(2) X{j-1}(2)],'c');

end

plot(x(1),x(2),'rx','LineWidth',2,'MarkerSize',8);

if mode==2

x\_new=varargin{1};

plot([x(1) x\_new(1)],[x(2) x\_new(2)],'m');

plot(x\_new(1),x\_new(2),'gx','LineWidth',2,'MarkerSize', 8);

elseif mode==3

plot(x(1),x(2),'ro','LineWidth',2,'MarkerSize',12);

end

hold off;

pause(0.1);

\*\*\*\*\*\*\*\*rnd\_search.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [x,f]=rnd\_search(fun,x,B,CData,max\_iter,tol\_x)

X{1}=x;

f=feval(fun,x);

range=0.2\*norm(B.ub-B.lb);

count=1;

while count<max\_iter & norm(range)>=tol\_x

step=(B.ub-B.lb).\*(rand(2,1)-0.5)\*range;

x\_new=max(min(x+step,B.ub),B.lb);

f\_new=feval(fun,x\_new);

rnd\_plot(CData,X,x,2,x\_new);

if f\_new<f

range=range\*2.0;

x=x\_new; f=f\_new;

X{length(X)+1}=x;

rnd\_plot(CData,X,x,1);

else

range=range/1.3;

end

count=count+1;

end

rnd\_plot(CData,X,x,3);

\*\*\*\*\*\*\*\*two\_peaks.m\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function z=two\_peaks(x)

z=(((x(1)-0.5).^2+(x(2)-0.5).^2).\*((x(1)+0.5).^2+(x(2)+0.5).^2)).^(0.5);

end