**MATLAB**

**Name:** Priyadarshi Roy

**Class:** CMSA

**Semester:** 6

**Roll No. :** 501

1.

**% Average**

function avg=average(a,n)

avg=sum1(a,n)/n;

end

2.

% Bubble sort

function a=bubble(a,n)

for i=1:n-2

for j=1:n-2-i

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

t=a(j);

a(j)=a(j+1);

a(j+1)=t;

end

end

end

end

3.

%Copying contents of one file to another

%METHOD 1

clc;

close all;

clear all;

file1 = input('\nEnter Input File Name=','s');

file2 = input('\nEnter Output File Name=','s');

fp1 = fopen(file1,'r');

fp2 = fopen(file2,'w');

fseek(fp1,0,1);

n = ftell(fp1);

fseek(fp1,0,-1);

while ~feof(fp1)

ch = fread(fp1,1);

fwrite(fp2,ch);

end

%METHOD 2

ch = fpread(fp1);%Reading enitre file in matrix ch

fwrite(fp2,ch);%Writing entire file in output file

[r,c] = size(ch);%n = size of input matrix

fprintf('\nSize of input file=%d\n',r);

4.

%function ce=expx(x) : Function to calculate exp(x) using Taylor Series

%expansion

function ce=expx(x)

format long;

ce=1;

pe=0;

eps=abs(ce-pe);

n=1;

fact=1;

px=1;

while eps>1e-06

pe=ce;

fact=fact\*n;

px=px^x;

ce=ce+px/fact;

eps=abs(ce-pe);

n=n+1;

end

end

5.

% Extract all handwritten characters

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

out\_file = ['aa.jpg';'ab.jpg';'ac.jpg';'ad.jpg';'ae.jpg';'af.jpg';'ag.jpg';'ah.jpg';'ai.jpg';'aj.jpg';

'ak.jpg';'al.jpg';'am.jpg';'an.jpg';'ao.jpg';'ap.jpg';'aq.jpg';'ar.jpg';'as.jpg';'at.jpg';'au.jpg';

'av.jpg';'aw.jpg';'ax.jpg';'ay.jpg';'az.jpg'];

x = imread(file1);

[r, c, d] = size(x);

%To convert RGB to Gray scale

g(1:r, 1:c, 1) = 0.2989\*x(1:r, 1:c, 1) + 0.587\*x(1:r, 1:c, 2) + 0.114\*x(1:r, 1:c, 3);

% To convert Gray Scale to binary image

for i=1:r

for j=1:c

if g(i,j,1)>=127

b(i,j,1) = 255;

else

b(i,j,1) = 0;

end

end

end

fprintf('\nr=%d c=%d d=%d\n',r,c,d);

imshow(b);

%To extract chracaters from input text file

n=0;

%n=counter for each character

c1 = 1;

cloop=1;

while cloop<=c

%To find left edge of any character

for i=c1:c

flag1=0;

%flag1=0 means we have obtained a white pixel

for j=1:r

if b(j,i,1)==0

c1 = i;

flag1 = 1;

break;

end

end

if flag1==1

break;

end

end

%To get right edge of a character

c2 = c1 + 1;

for i=c2:c

flag2=1;

for j=1:r

if b(j,i,1)==0

flag2=0;

break;

end

end

if flag2==1

c2 = i - 1;

break;

end

end

%To extract a character from i=c1 to i=c2

y1(1:r, 1:c2-c1+1, 1) = 255;

if flag1==1 && flag2==1

y1(1:r, 1:c2-c1+1, 1) = b(1:r, c1:c2, 1);

n = n+1;

fprintf('c1=%d c2=%d n=%d\n',c1,c2,n);

imwrite(y1, out\_file(n,:), 'jpg');

cloop = cloop + c2 - c1 + 1;

c1 = c2 + 1;

end

end

%End of program

6.

% Hanoi

function n = hanoi(p1,p2,p3,nd)

global n;

if nd==1

n=n+1;

fprintf('Move Disk-%d from Peg-%d\n',nd,p1,p2);

return;

end

hanoi(p1,p3,p2,nd-1);

n=n+1;

fprintf('Move Disk-%d from Peg-%d to Peg-%d\n',nd,p1,p2);

hanoi(p3,p2,p1,nd-1);

end

%Implementation of hanoi

clc;

close all;

clear all;

global n;

nd=input('\nEnter number of disks to be moved=');

n=0;

hanoi(1,2,3,nd);

fprintf('\nTotal number of operations performed=%d\n',n);

7.

% image\_proc\_1.m: Write a program to display an image on screen

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

file2 = input('\nEnter Output Image File Name=','s');

x = imread(file1);

[r,c,d] = size(x);

fprintf('r=%d\nc=%d\nd=%d\n',r,c,d);

imshow(x);

imshow(x, file2);

%End of program

8.

% image\_proc\_2x.m: Write a program to split 1 image to 2 images.

% Split along x-axis.

clc;

clear all;

close all;

file1 = input('\nEnter Input Image file name=','s');

file2 = input('\nEnter Output Image file-1 name=','s');

file3 = input('\nEnter Output Image file-2 name=','s');

x = imread(file1);

[r,c,d] = size(x);

r1 = fix(r/2);

y1(1:r1,1:c,1:d) = x(1:r1,1:c,1:d);

y2(1:r-r1,1:c,1:d) = x(r1+1:r,1:c,1:d);

imwrite(y1, file2);

imwrite(y2, file3);

imshow(y1);

%end of program

9.

% image\_proc\_2.m: Write a program to split 1 image to 2 images.

% Split along y-axis.

clc;

clear all;

close all;

file1 = input('\nEnter Input Image file name=','s');

file2 = input('\nEnter Output Image file-1 name=','s');

file3 = input('\nEnter Output Image file-2 name=','s');

x = imread(file1);

[r,c,d] = size(x);

c1 = fix(c/2);

y1(1:r,1:c1,1:d) = x(1:r,1:c1,1:d);

y2(1:r,1:c-c1,1:d) = x(1:r,c1+1:c,1:d);

imwrite(y1, file2);

imwrite(y2, file3);

imshow(y1);

%end of program

10.

%Write a program to copy an image n-times along x-axis

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

file2 = input('\nEnter Output Image File Name=','s');

nx = input('Enter number of times images to be copied along x-axis=');

x = imread(file1);

[r, c, d] = size(x);

%To copy image along x-axis

c1 = 1;

for i=1:nx

y(1:r, c1:i\*c, 1:d) = x(1:r, 1:c, 1:d);

c1 = c1 + c;

end

imshow(y);

imwrite(y, file2);

%End of program

11.

%Write a program to copy an image n-times along x-axis

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

file2 = input('\nEnter Output Image File Name=','s');

nx = input('Enter number of times images to be copied along both-axis=');

x = imread(file1);

[r, c, d] = size(x);

%To copy image along x-axis

c1 = 1;

for i=1:nx

y(1:r, c1:i\*c, 1:d) = x(1:r, 1:c, 1:d);

c1 = c1 + c;

end

[r, c, d] = size(y);

r1 = 1;

for i=1:nx

z(r1:i\*r, 1:c, 1:d) = y(1:r, 1:c, 1:d);

r1 = r1 + r;

end

imshow(z);

imwrite(z, file2);

%End of program

12.

%Write a program to copy an image n-times along y-axis

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

file2 = input('\nEnter Output Image File Name=','s');

nx = input('Enter number of times images to be copied along y-axis=');

x = imread(file1);

[r, c, d] = size(x);

%To copy image along y-axis

r1 = 1;

for i=1:nx

y(r1:i\*r, 1:c, 1:d) = x(1:r, 1:c, 1:d);

r1 = r1 + r;

end

imshow(y);

imwrite(y, file2);

%End of program

13.

% largest

function max=largest(a,n)

max=a(1);

for i=1:n

if a(i)>max

max=a(i);

end

end

end

14.

%main

%clc;

%clear all;

%close all;

s = sum1(a,n);

avg = average(a,n);

sdev = std\_dev(a,n);

l = largest(a,n);

fprintf('Sum = %5.5f\nAverage = %5.5f\nLargest = %5.5f\nStandar Deviation = %5.5f',s,avg,sdev,l);

15.

%rung4\_main.m : Write main program to access rung4 function

clc;

clear all;

close all;

x0=input('\nEnter x0=');

y0=input('Enter y0=');

xn=input('Enter xn=');

h=input('Enter h=');

n=fix((xn-x0)/h);

x(1)=x0;

y(1)=y0;

for i=2:n+1

x(i)=x0+h;

y1=runge4(x0,y0,h);

y(i)=y1;

x0=x(i);

y0=y1;

end

for i=1:n+1

fprintf('x(%d)=%5.1f y(%d)=%7.5f\n',i,x(i),i,y(i));

end

plot(y,'bo-');

%end of main

16.

%pi using Taylor Series

function cp=pie()

cp=1;

pp=0;

eps=abs(cp-pp);

n=3;

sf=-1;

p3=1;

while eps>1e-16

pp=cp;

p3=p3\*3;

cp=cp+sf/(p3\*n);

eps=abs(cp-pp);

n=n+2;

sf=-sf;

end

cp=sqrt(3)\*2\*cp;

end

17.

clc;

close all;

clear all;

a=input('\nEnter 1st number=');

b=input('\nEnter 2nd number=');

c=input('\nEnter 3rd number=');

big=a;

if b><big

big=b

end

if c>big

big=c

end

fprintf('Largest = '+big);

18.

%program\_1.m: Write a program to calculate (i) number of vowels

%(ii) number of consonants (iii) number of alphabets in any text file

clc;

close all;

clear all;

file1 = input('\nEnter Input File Name=','s');

nv=0;

nc=0;

na=0;

fp1=fopen(file1,'r');

while ~feof(fp1);

ch = fread(fp1, 1, 'uint8=>char');

fprintf('%c',ch);

ch1 = upper(ch);

if ch1>='A' & ch1<='Z'

na=na+1;

if ch1=='A' || ch1=='E' || ch1=='I' || ch1=='O' || ch1=='U'

nv=nv+1;

end

end

end

nc=na-nv;

fprintf('\nNumber of Alphabets= %d\nNumber of Vowels= %d\nNumber of consonants= %d ',na,nv,nc);

fclose('all');

%end of program

19.

clc;

close all;

clear all;

n1=input('\nEnter value of n (1-20)=');

for i = 1 : n1

r(i)=input('\nEnter roll number =');

na(i,:)=input('Enter name=','s');

m(i)=input('Enter marks =');

end

% Bubble sort algorithm starts

pass=0;

flag=0;

n=n1;

while flag==0

flag=1;

pass=pass+1;

n=n-1;

for i=1:n

if na(i)>na(i+1)

t=na(i);

na(i)=na(i+1);

na(i+1)=t;

t=r(i);

r(i)=r(i+1);

r(i+1)=t;

t=m(i);

m(i)=m(i+1);

m(i+1)=t;

flag=0;

end

clc;

fprintf('\nPass=%d flag=%d Step=%d\n',pass,flag,i);

fprintf('\nRoll\tName\tMarks\n');

for j=1:n1

fprintf('\n%d\t\t%s\t\t%d',r(j),na(j,:),m(j));

end

choice=input('\nPress enter key to continue--->','s');

end

end

20.

%stego1.m : Write a program to extract bits from any given file

clc;

close all;

clear all;

file1=input('\nEnter Input File Name=','s');

file2=input('Enter Output bit File name=','s');

file3=input('Enter Output stego File Name=','s');

str=input('\nEnter any secret message=','s');

fp1=fopen(file1,'r');

fp2=fopen(file2,'w');

while ~feof(fp1)

ch=fscanf(fp1,'%c',1);

ch1=cast(ch,'uint8');

fprintf('\n%c :',ch);

for i=8:-1:1

b=bitget(ch1,i);

fprintf('%d',b);

fprintf(fp2,'%d',b);

end

end

% To convert String to bits

m=length(str);

j=0;

fprintf('\nSecret Message--->\n')

for i=1:m

ch=str(i);

ch1=cast(ch,'uint8');

fprintf('\n%c :',ch);

for k=8:-1:1

j=j+1;

b(j)=bitget(ch1,k);

fprintf('%d',b(j));

end

end

fclose(fp1);

fclose(fp2);

21.

% reflection.m : Write a program to take reflection of an image along

% x-axis.

clc;

close all;

clear all;

file1 = input('\nInput File Name=','s');

file2 = input('\nEnter Output Image Name=','s');

x = imread(file1);

[r,c,d] = size(x);

y=x;

y(r+1:2\*r,1:c,1:d) = x(r:-1:1,1:c,1:d);

imwrite(y,file2);

imshow(y);

fclose('all');

%end of program

22.

% pattern\_replace.m: Write a program to replace old pattern by a new

% pattern

clc;

close all;

clear all;

file1 = input('\nEnter Input File Name=','s');

file2 = input('\nEnter Output File Name=','s');

opat = input('\nEnter old pattern to be searched=','s');

npat = input('\nEnter new pattern to be replaced=','s');

fp1 = fopen(file1,'r');

fp2 = fopen(file2,'w');

leno = length(opat);

lenn = length(npat);

nc=0;

while ~feof(fp1)

ch = fread(fp1, 1, 'uint8=>char');

ch1 = upper(ch);

if ch1==upper(opat(1))

flag=1;

for i=1:leno-1

j=i+1;

ch=fread(fp1, 1, 'uint8=>char');

if upper(ch) ~= upper(opat(j))

flag=0;

break;

end

end

if flag==1

for i1=1:lenn

fwrite(fp2, npat(i1));

end

nc=nc+1;

else

fseek(fp1, -j, 0);

for i1=1:j

ch = fread(fp1, 1, 'uint8=>char');

fwrite(fp2, ch);

end

end

else

fwrite(fp2, ch);

end

end

fclose('all');

fprintf('\nNumber of times <%s> is replaced by <%s> = %d',opat,npat,nc);

%End of program

23.

% Write a program to convert any RGB image (i) Gray Scale

% and (ii) B/W image file

clc;

close all;

clear all;

file1 = input('\nEnter Input Image File Name=','s');

file2 = input('\nEnter Grayscale Output Image File Name=','s');

file3 = input('\nEnter B/W Output Image File Name=','s');

x = imread(file1);

[r,c,d] = size(x);

% To convert RGB to Gray Scale

g(1:r, 1:c, 1) = 0.2989\*x(1:r,1:c,1) + 0.587\*x(1:r, 1:c, 2) + 0.114\*x(1:r, 1:c, 3);

imwrite(g, file2);

%Saving gray scale image in file2

%To convert Gray scale image to B/W Image

for i=1:r

for j=1:c

if g(i,j,1)>=127

b(i,j,1) = 255;

else

b(i,j,1) = 0;

end

end

end

%Saving Binary Image file in file3

imwrite(b, file3);

imshow(b);

%End of program

24.

% Write a program to convert any RGB image to GRAY scale image.

clc;

close all;

clear all;

file1 = input('\nInput File Name=','s');

file2 = input('\nEnter Output Image Name=','s');

x = imread(file1);

[r,c,d] = size(x);

%Black and white

g(1:r,1:c,1) = 0.2989\*x(1:r,1:c,1) + 0.587\*x(1:r,1:c,2) + 0.1140\*x(1:r,1:c,3);

%copying to RGB

y = x;

y(1:r,c+1:2\*c,1) = g(1:r,1:c,1);

y(1:r,c+1:2\*c,2) = g(1:r,1:c,1);

y(1:r,c+1:2\*c,3) = g(1:r,1:c,1);

imwrite(y, file2);

imshow(y);

%end

25.

%function [y]=runge4(x0,y0,h) : Function to solve differential equation

%using 4-th order runge kutta method

function [y]=runge4(x0,y0,h)

k1=h\*y0;

k2=h\*(y0+k1/2);

k3=h\*(y0+k2/2);

k4=h\*(y0+k3);

y=y0+1/6\*(k1+2\*k2+2\*k3+k4);

end

26.

% Small capital to small letters

clc;

close all;

clear all;

file1 = input('\nEnter File Name=','s');

fp1 = fopen(file1,'r+');

fseek(fp1,0,1);%Moving file pointer to end of file

n = ftell(fp1);%n = Size of input file

fseek(fp1,0,-1);%Moving file pointer to beginninfg of file

nc=0;% number of bytes modified

for i=1:n

%ch=fread(fp1,1,'uint8=>char')%character wise reading

ch = fread(fp1,1)%int wise reading

if ch>=65 && ch<=90

nc=nc+1;

fseek(fp1,-i,0);

ch=ch+32;

fwrite(fp1,ch);

end

end

fclose('all');

fprintf('\nSize of <%s>=%d Bytes and number of Bytes and number of Bytes modified=%d Bytes \n',file1,n,nc);

%END

27.

% Split one file into two files

clc;

close all;

clear all;

file1 = input('\nEnter Input File Name=','s');

file2 = input('\nEnter Output File Name 1=','s');

file3 = input('\nEnter Output File Name 2=','s');

fp1 = fopen(file1,'r');

fp2 = fopen(file2,'w');

fp3 = fopen(file3,'w');

fseek(fp1,0,1);%Moving file pointer to end of file

n = ftell(fp1);%n = Number of bytes in input file

n1 = fix(n/2);

fseek(fp1,0,1);%Moving file pointer to beginning of input file

% To split input file into 2 output files

for i=1:n

ch=fread(fp1,1);

if i<=n1

fwrite(fp2,ch);

else

fwrite(fp3,ch);

end

end

fclose('all');

fprintf('\nFile split is over\n');

fprintf('\nSize of <%s> = %d Bytes',file2,n1);

fprintf('\nSize of <%s> = %d Bytes',file3,(n-n1));

fprintf('\nSize of <%s> = %d Bytes',file1,n);

%END

28.

% standard deviation

function sd=std\_dev(a,n)

s=0;

avg=average(a,n);

for i=1:n

s=s+((a(i)-avg)^2);

end

s=s/n;

sd=sqrt(s);

end

29.

function s=sum1(a,n)

s=0;

for i=1:n

s=s+a(i);

end

end

30.

% Calculate number of words in a file

clc;

close all;

clear all;

file1 = input('\nEnter Input file name:','s');

fp1 = fopen(fp1,'r');

s=''

while ~feof(fp1)

ch = fpread(fp1,1,'unit8=>char');

if s<>' '

s=s+ch;

else

fprint('%s\n',s);

s='';

31.

% Running bubble sort

s = bubble(a,n);

fprintf('Sorted list:');

fprintf('%5.1f',s);

32.

%Write a program to obtain exp(x) from x=0 to 2 in step of .1.

clc;

clear all;

close all;

for x=0:.1:2

y=expx(x);

fprintf('exp(%5.1f)=%16.12f\n',x,y);

end