1. function f=eval(x)

f=(x.^4\*sqrt(3\*x+5))/(x.^2+1).^2;

end

>> value=eval(6)

value =

4.5401

1. function [average,deviation] = avgdev (x)

average=sum(x)/length(x)

deviation=std(x)

end

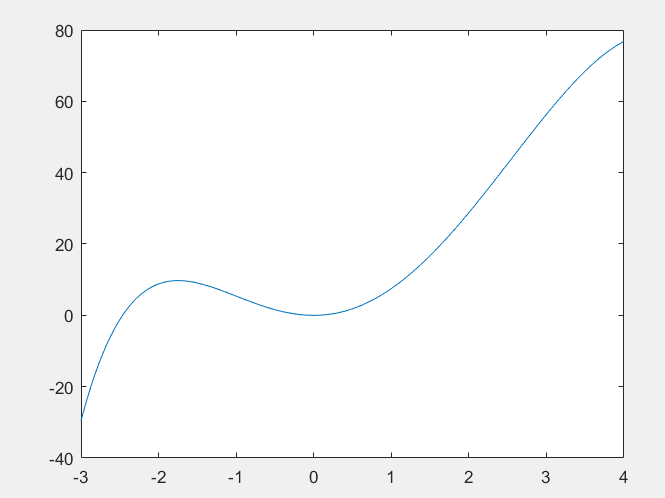
>> avgdev([80,75,91,60,79,89,65,80,95,50,81])

average =

76.8182

deviation =

13.6661



3)

function value = y(x)

value=(-0.2.\*x.^4)+(2.^(-0.5.\*x).\*x.^3)+7.\*x.^2

end

>> y(-2.5)

ans =

-1.2252

>> y(3)

ans =

56.3459

>> x=-3:0.01:4;

>> plot(x,y(x))

4)

function [root] = Roots(p)

root = roots(p);

end

>> y=input('Enter a polynomial function like a vector ex:[3 2 1]: ');

>> Roots(y)

For [3 2 1]

ans =

-0.3333 + 0.4714i

-0.3333 - 0.4714i

5)

function OddEven(x)

if rem(x, 2) == 0

disp ('The number is even');

else

disp ('The number is odd');

end

>> OddEven(7)

The number is odd

6)

function Menu

m = input('Enter first number ?');

n = input('Enter second number ?');

disp('1)Addition');

disp('2)Subtraction');

disp('3)Multiplication');

disp('4)Division');

menu=input('Enter your choice ?');

switch(menu)

case 1

m+n

case 2

m-n

case 3

m\*n

case 4

m/n

end

>> Menu

Enter first number ?1

Enter second number ?2

1)Addition

2)Subtraction

3)Multiplication

4)Division

Enter your choice ?1

ans =

3

7)

function PosNeg(x)

if x == 0

disp ('equal to zero');

elseif x > 0

disp ('positive');

else

disp ('negative');

end

>> PosNeg(4)

positive

>> PosNeg(-1)

negative

8)

function Kelvin()

kelvin = input('Enter the temperature in Kelvin :');

degree=kelvin-273.15

if degree < 32

disp ('ICE');

elseif degree < 212

disp ('WATER');

else

disp ('STEAM');

end

>> Kelvin

Enter the temperature in Kelvin :273

degree =

-0.1500

ICE

9)

function Prime

n = input('Enter the number :');

i=2;

while i<=sqrt(n)

if n==0 | n==1

disp('not a prime number');

elseif rem(n,i)==0

disp('not a prime number');

break

end

i=i+1;

end

if i>sqrt(n)

disp('is a prime number');

end

end

>> Prime

Enter the number :12

not a prime number

>> Prime

Enter the number :3

is a prime number

10)

function Thirteen()

for i=1:13

disp(13\*i)

end

>> Thirteen

13

26

39

52

65

78

91

104

117

130

143

156

169

11)

>> char(32+10\*tril(ones(4)))

ans =

\*

\*\*

\*\*\*

\*\*\*\*

12)

function starpattern2

for i = 1:3

for j = 1:3-i

fprintf(' ');

end

for j = 1:(2\*i)-1

fprintf('\*');

end

fprintf('\n');

end

for i = 2:-1:1

for j = 1:3-i

fprintf(' ');

end

for j = 1:(2\*i)-1

fprintf('\*');

end

fprintf('\n');

end

>> starpattern2

\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*

13)

function factorial(num)

fact = 1;

if num < 0

disp('factorial does not exist for negative numbers')

elseif num == 0

disp('The factorial of 0 is 1')

else

for i=1:num

fact = fact\*i;

end

disp(fact)

disp('is the factorial')

end

end

>> factorial(5)

120

is the factorial

14)

function fibonacci

a(1) = 0;

a(2) = 1;

n = input('Enter number of term desired ');

for i = 3:n

a(i) = a(i-1)+a(i-2);

end

for i = 1:n

disp(a(i))

end

>> fibonacci

Enter number of term desired 5

0

1

1

2

3

15)

function sumsquares

list=2:1:20;

list2=power(list,2);

sum(list2)

>> sumsquares

ans =

2869

16)

function oddint

list=1:2:501;

sumval=sum(list)

>> oddint

sumval =

63001

17)

years = 0:30;

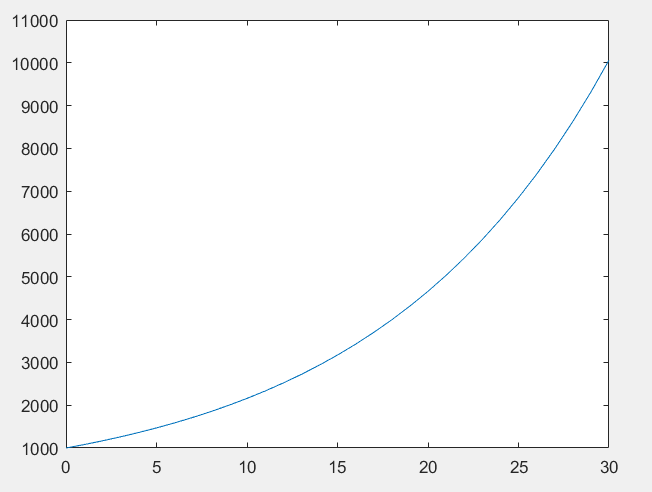
account\_balance = [1000];

for i = 1:30

account\_balance(end+1) = account\_balance(end)+ (account\_balance(end)\*(2/25));

end

plot(years, account\_balance)



18)years = 0:30;

account\_balance = [5000];

count = 0;

while account\_balance(end) < 1000000

account\_balance(end+1) = account\_balance(end)+ (account\_balance(end)\*(9/100));

count = count+1;

end

fprintf('Number of years it takes to reach retirement is: %d\n', count)

>> oddint

Number of years it takes to reach retirement is: 62

19)

function mps = mphTOmets(mph)

mps=mph\*0.44704;

end

>> mphTOmets(55)

ans =

24.5872

20)

function [Area]= triangle(a,b,c)

s=(a+b+c)./3;

Area=sqrt(s.\*(s-a).\*(s-b).\*(s-c))

>> triangle(3, 8, 10)

Area =

9.1652

>> triangle(7, 7, 5)

Area =

1.9373

21)

function [ g ] = fgrade( R )

g = zeros(size(R, 1), 1);

for r = 1:size(R,1)

hw\_grades = R(r, 1:6);

drop = min(hw\_grades);

drop\_index = find(hw\_grades == drop);

drop\_index = drop\_index(1);

hw\_grades = hw\_grades([1:drop\_index-1 drop\_index+1:end]);

hw\_avg = mean(hw\_grades);

midterm\_avg = mean(R(r, 7:9));

final = R(r, 10);

if midterm\_avg > final

g(r, 1) = 0.85\*midterm\_avg + 0.15\*hw\_avg;

else

g(r, 1) = 0.15\*R(r, 7) + 0.15\*R(r, 8) + 0.15\*R(r, 9) + 0.15\*midterm\_avg + 0.15\*hw\_avg;

end

end

end

%fgrade([8, 9, 6, 10, 9, 7, 76, 86, 91, 80])

%ans =72.9733

%fgrade([7, 10, 6, 9, 10, 9, 91, 71, 81, 88])

%ans =49.9500

%fgrade([5, 5, 6, 1, 8, 6, 59, 72, 66, 59])

%ans =56.7167

%fgrade([6, 8, 10, 4, 5, 9, 72, 78, 84 78])

%ans =47.9400

%fgrade([7, 7, 8, 8, 9, 8, 83, 82, 81 84])

%ans =50.4000

22)

function series

i = 1;

sum = 0;

sign = 1;

while i <= 9999

sum = sum + sign \* (i\*i);

sign = sign \* -1;

i = i + 2;

end

fprintf('%d\n',sum)

end

>> series

-50000000

23)

function square\_root(n)

for i = 2:2:n

fprintf('%d\n',sqrt(i));

end

end

>> square\_root(4)

1.414214e+00

2

24)

function ifprime

for i = 1:1000;

if isprime(i)

fprintf('%d\n', i)

end

end

>> ifprime

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

53

59

61

67

71

73

79

83

89

97

101

103

107

109

113

127

131

137

139

149

151

157

163

167

173

179

181

191

193

197

199

211

223

227

229

233

239

241

251

257

263

269

271

277

281

283

293

307

311

313

317

331

337

347

349

353

359

367

373

379

383

389

397

401

409

419

421

431

433

439

443

449

457

461

463

467

479

487

491

499

503

509

521

523

541

547

557

563

569

571

577

587

593

599

601

607

613

617

619

631

641

643

647

653

659

661

673

677

683

691

701

709

719

727

733

739

743

751

757

761

769

773

787

797

809

811

821

823

827

829

839

853

857

859

863

877

881

883

887

907

911

919

929

937

941

947

953

967

971

977

983

991

997

25)

function compositesum

sum = 0;

for i = 1:50;

if ~isprime(i)

sum = sum + i;

end

end

fprintf('The sum is %d\n', sum)

>> compositesum

The sum is 947

26)

function randomnumber

flag = 0;

while flag == 0

rand\_num = randi([3 10], 1, 1);

if rand\_num == 5

fprintf('5 Occured !!\n')

flag = 1;

else

fprintf('%d occured but not 5\n', rand\_num)

end

end

>> randomnumber

9 occured but not 5

10 occured but not 5

4 occured but not 5

10 occured but not 5

8 occured but not 5

3 occured but not 5

5 Occured !!

27)

function randarray

count = 1;

array = zeros(1, 10);

while count < 11

rand\_num = randi([3 10], 1, 1);

array(count) = rand\_num;

count = count + 1;

end

display(array);

>> randarray

array =

7 10 10 4 10 10 6 9 4 6