**IF, SWICH**

1. Napisati skript koji obračunava platu radnika. Korisnik unosi broj radnih sati i zaradu po satu. Preko 40 sati rada nedeljno se plaća 50 procenata vise.

Resenje:

t = input('Unesite broj radnih sati ');

h = input ('Unesite zaradu po satu sata ');

plata = t\*h;

if t>40

 plata = plata + (t-40)\*0.5\*h;

end

fprintf('Ukupna plata je %5.2f dinara',plata);

2. Napisati program u skriptu koji odredjuje realne korene kvadratne jednačine ax^2+bx+c=0. Korisnik unosi a, b i c. Najpre se izračunava diskriminanta i za D<0 ispisuje se poruka ‘Nema realnih korena’, za D=0 poruka ‘Jednacina ima jedan dvostruki koren jednak:’ i ispisuje vrednost korena u sledecem redu i za D>0 ispisuje se poruka ‘Jednacina ima dva realna korena:’ i ispisuje korene u sledećem redu.

Resenje:

a=input('unesi a');

b=input('unesi b');

c=input('unesi c');

D=b^2-4\*a\*c;

if D>0

 disp('ima dva realna resenja');

 x1=(-b+sqrt(D))/(2\*a);

 x2=(-b-sqrt(D))/(2\*a);

 fprintf('resenja su %f i %f',x1,x2);

elseif D<0

 disp('nema realna resenja');

 else

 disp('ima dvostruki realni koren')

 x=(-b)/(2\*a);

 fprintf('resenje je %f',x);

end

3. Napisati skript koji obračunava cenu vožnje taksijem to tako što je cena prvog kilometra 200 dinara. Svaki sledeći kilometar do 10 kilometara je 25 dinara. Svaki kilometar preko 10 km je 10 dinara. Kilometri se zaokružuju na najbliži ceo broj, osim za prvi kilometar, gde mora biti plaćen bar 1 km vožnje (start). Deca mladja od 18 godina i stariji od 60 godina dobijaju popust 20 %. Korisnik unosi uzrast i dužinu putovanja, a kao informaciju dobija koliko će ga koštati vožnja.

Resenje:

duzina = input('Unesite duzinu voznje ');

 uzrast = input('Unesite uzrast ');

 trosak = 200;

 duzina = round(duzina);

 if duzina <= 10

 trosak = trosak + max([0 (duzina-1)]) \* 25;

 else

 trosak = trosak + 9 \* 25 + (duzina-10) \* 10;

 end

 if uzrast <= 18 || uzrast >= 60

 trosak = trosak \* 0.8;

 end

fprintf('Cena vase voznje iznosi %5.2f dinara \n', trosak)

4. Koristeći switch naredbu napisati skript koji unetu duzinu u km, m, dm,cm prebacuje u mm, tako što korisnik unosi jedinicu mere i dužinu u toj jedinici, a skript vraća odgovor kolika je dužina u mm.

Resenje:

jedinica = input('Unesite jedinicu u kojoj je zadata duzina ','s');

duzina = input('Unesite duzinu ');

switch jedinica

 case 'cm'

 rezultat = duzina \* 10 ;

 case 'dm'

 rezultat = duzina \* 100 ;

 case 'm'

 rezultat = duzina \* 1000 ;

 case 'km'

 rezultat = duzina \* 1000000 ;

end

fprintf('Duzina u milimetrima je %10.0f \n', rezultat);

5. Napisati skript koji sortira u rastućem redosledu niz od tri elementa koji unosi korisnik, ne koristeći naredbu sort.

Resenje:

v = input('Unesite vektor od tri elementa: ');

if v(2) < v(1)

 v = [v(2) v(1) v(3)];

 end

 if v(3) < v(2)

 v = [v(1) v(3) v(2)];

 end

 if v(2) < v(1)

 v = [v(2) v(1) v(3)];

 end

 fprintf('Sortiran vektor je: \n');

 disp(v)

6. Napisati skript u kome korisnik unosi redni broj dana u nedelji, počev od nedelje kao prvog dana i zatim se koristeći if elseif naredbu ispisuje poruka koji dan je u pitanju i da li je radni dan ili vikend.

Na primer ako korisnik unese 2 treba da dobije poruku „Ponedeljak, sto je radni dan.“, ako unese 7 „Subota, sto je neradni dan.“, a ukoliko ne unese broj između 1 i 7 poruku „Broj mora biti od 1 do 7.“

Resenje:

n = input('Unesite redni broj dana: ');

if n == 1

 fprintf('Nedelja,');

 day\_type = 2;

elseif n == 2

 fprintf('Ponedeljak,');

 day\_type = 1;

elseif n == 3

 fprintf('Utorak,');

 day\_type = 1;

elseif n == 4

 fprintf('Sreda,');

 day\_type = 1;

elseif n == 5

 fprintf('Cetvrtak,');

 day\_type = 1;

elseif n == 6

 fprintf('Petak,');

 day\_type = 1;

elseif n == 7

 fprintf('Subota,');

 day\_type = 2;

else

 fprintf('Broj mora biti od 1 do 7.\n');

 return

end

if day\_type == 1

 fprintf(' sto je radni dan. \n');

else

 fprintf(' sto je neradni dan. \n');

end

Napomena: Zadatak je bolje resavati sa switch case naredbom.

**PETLJE**

1. Izračunati sumu $\sum\_{k=1}^{n}\frac{(-1)\^k}{2^{k}}k $sa for petljom i bez petlje, gde n unosi korisnik.

Resenje:

n = input('Unesite broj n ');

S = 0;

for k=1:n

 S=S+(-1)^k\*k/2^k;

end

fprintf('Suma je %f \n', S);

% bez petlje

n = input ('Unesite n ');

k = 1:n;

result = sum(((-1).^k.\*k)./ (2.^k));

disp(result);

1. Napisati program u skriptu koji odredjuje i ispisuje n-ti Fibonacijev broj, gde n unosi korisnik.

Resenje:

n = input('Unesite prirodan broj n: ');

f(1) = 1;

f(2) = 1;

for i = 3:n

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

end

fprintf('%i-ti Fibonacijev broj je %i \n', n,f(n))

1. Koristeći for petlju napisati program u skriptu koji iscrtava grafike funkcije f(x)=cos(nx) za n = 1,2,3,4,5,6 i 0 <=x<=2\*Pi.

Resenje:

figure

hold on

x = linspace(0,2\*pi);

for n = 1:6

 subplot(3,2,n);

 y = cos(n\*x);

 plot(x,y);

 axis tight

 title('Grafik funkcije cos(',num2str(n),'x)') % pogledajte HELP num2srt

end

1. Korisnik unosi dva niza x i y. Koristeći for petlju kreirati matricu A čiji elementi su definisani kao $A\_{ij}=x\_{i}y\_{j} $.

Resenje:

x = input('Unesite prvi niz: ');

y = input('Unesite drugi niz: ');

n = length(x);

m = length(y);

for i = 1:n

 for j = 1:m

 A(i,j) = x(i)\*y(j);

 end

end

disp('Trazena matrica je: ')

A

1. Koristeći Tejlorov polinom funkcije sin(x)

$\sum\_{k=0}^{\infty }\frac{(-1)^{k}x^{2k+1}}{\left(2k+1\right)!}$ izračunati sa tačnošću od 0.0001 vrednost sin(x) gde je x vrednost ugla u stepenima koji unosi korisnik.

Resenje:

x = input('Unesite vrednost ugla u stepenima: ');

xr = x\*pi/180;

sabirak = 1;

suma = 0;

k = 0;

while abs(sabirak) > 0.00001

 sabirak = (-1)^k/factorial(2\*k+1)\*xr^(2\*k+1);

 suma = suma + sabirak;

 k = k+1;

end

fprintf('Rezultat je: %f \n' , suma);

fprintf('Postignut je u %i koraka.\n', k);

1. Napisati program koji odredjuje kusur koji kasa za samousluživanje treba da vrati korisniku.

Program generiše cenu kao slučajanu celobrojnu veličinu manju od 2000. Zatim mašina pita korisnika kojom novčanicom plaća

– i to mora biti jedna novčanica iznosa 2000, 1000, 500, 200, 100,50, 20, 10, 5, 2 ili 1 dinar. Ako je uneta novčanica manja od cene

ispisuje se odgovarajuća poruka. Ako je novčanica dovoljna, program izračunava kusur i ispisuje kojim novčanicama se kusur isplaćuje,

tako da se upotrebi najmanji mogući broj novčanica.

Resenje:

price = randi(2000);

fprintf ('Treba platiti %i \n', price);

payment = input ('Unesite sa kojom novacanicom placate ');

if payment < price

 display ('Greska, niste uneli dovoljno novca.');

 break;

end;

change = payment-price;

while change > 0

 if (change >= 1000)

 display ('novacanica od 1000 ');

 change = change - 1000;

 continue;

 end;

 if (change >= 500)

 display ('novacanica od 500 ');

 change = change - 500;

 continue;

 end;

 if (change >= 100)

 display ('novacanica od 100 ');

 change = change - 100;

 continue;

 end;

 if (change >= 50)

 display ('novacanica od 50 ');

 change = change - 50;

 continue;

 end;

 if (change >= 20)

 display ('novacanica od 20 ');

 change = change - 20;

 continue;

 end;

 if (change >= 10)

 display ('novacanica od 10 ');

 change = change - 10;

 continue;

 end;

 if (change >= 5)

 display ('novacanica od 5 ');

 change = change - 5;

 continue;

 end;

 if (change >= 2)

 display ('novacanica od 2 ');

 change = change - 2;

 continue;

 end;

 if (change >= 1)

 display ('novacanica od 1 ');

 change = change - 1;

 continue;

 end;

end;

Drugo resenje (resenje studenta):

cena = randi(2000);

fprintf('Iznos racuna je %i RSD\n', cena);

novac = input('Unesite apoen kojim placate racun: ');

kusur = novac - cena;

if kusur < 0

 disp('Niste uneli dovoljnu kolicinu novca');

elseif kusur == 0

 disp('Nema kusura');

else

 novcanice = zeros(1,11)

 while kusur > 0

 if kusur >= 2000

 novcanice(1) = novcanice(1) + 1;

 kusur = kusur - 2000;

 elseif kusur >= 1000

 novcanice(2) = novcanice(2) + 1;

 kusur = kusur - 1000;

 elseif kusur >= 500

 novcanice(3) = novcanice(3) + 1;

 kusur = kusur - 500;

 elseif kusur >= 200

 novcanice(4) = novcanice(4) + 1;

 kusur = kusur - 200;

 elseif kusur >= 100

 novcanice(5) = novcanice(5) + 1;

 kusur = kusur - 100;

 elseif kusur >= 50

 novcanice(6) = novcanice(6) + 1;

 kusur = kusur - 50;

 elseif kusur >= 20

 novcanice(7) = novcanice(7) + 1;

 kusur = kusur - 20;

 elseif kusur >= 10

 novcanice(8) = novcanice(8) + 1;

 kusur = kusur - 10;

 elseif kusur >= 5

 novcanice(9) = novcanice(9) + 1;

 kusur = kusur - 5;

 elseif kusur >= 2

 novcanice(10) = novcanice(10) + 1;

 kusur = kusur - 2;

 elseif kusur >= 1

 novcanice(11) = novcanice(11) + 1;

 kusur = kusur - 1;

 end

 end

 prikaz = [novcanice; 2000 1000 500 200 100 50 20 10 5 2 1];

 fprintf('Kusur iznosi %i RSD, sto je po apoenima:\n',(novac-cena));

 fprintf('%i x %i RSD\n',prikaz);

end

Trece resenje (resenje studenta)

clear

clc

cena= randi(2000)

novcanica = input('Kojom novcanicom placate racun? ');

if cena > novcanica

 fprintf('Novcanica nije dovoljna! Potrebna je veca od %i .\n', cena)

 return

end

kusur = novcanica - cena;

fprintf('Vas kusur koji trebate dobiti iznosi %i .\n', kusur)

k = 1;

n(1) = 0;

 while kusur >= 1000

 n(k) = 1000

 kusur = kusur - 1000 ;

 k = k + 1;

 end

 while kusur >= 500

 n(k) = 500;

 kusur = kusur - 500;

 k = k + 1;

 end

 while kusur >= 200

 n(k) = 200;

 kusur = kusur - 200;

 k = k + 1;

 end

 while kusur >= 100

 n(k) = 100;

 kusur = kusur - 100;

 k = k + 1;

 end

 while kusur >= 50

 n(k) = 50;

 kusur = kusur - 50;

 k = k + 1;

 end

 while kusur >= 20

 n(k) = 20;

 kusur = kusur - 20;

 k = k + 1;

 end

 while kusur >= 10

 n(k) = 10;

 kusur = kusur - 10;

 k = k + 1;

 end

 while kusur >= 5

 n(k) = 5;

 kusur = kusur - 5;

 k = k + 1;

 end

 while kusur >= 2

 n(k) = 2;

 kusur = kusur - 2;

 k = k + 1;

 end

 while kusur >= 1

 n(k) = 1;

 kusur = kusur - 1;

 k = k + 1;

 end

 disp('Vas kusur dobijate u novcanicama od: ')

 fprintf('%i dinara \n',n')

 clear n

 clear k

zadatak 3.2.2

MODELI PROSTORNE RASPODELE KRETANJA

Data je matrica razmene kretanja u postojecem stanju(Matrica t).

Na osnovu modela prosecnog faktora rasta prognozirati buducu razmenu kretanja izmedju zona. Kriterijum 0.95<F< 1.05 neka bude merodavan za broj iteracija.

t=[0 20 40; 50 0 15; 40 15 0];

T\_vrste = [90 80 85];

T\_kolone = [120 55 75];

[m,n] = size(t);

sumt\_vrste = [];

sumt\_vrste = sum(t')

sumt\_kolone = sum(t)

F\_vrste = T\_vrste./sumt\_vrste

F\_kolone = T\_kolone./sumt\_kolone

T1 = [];

while (((max(F\_vrste))>=1.05)&&((max(F\_kolone))>=1.05))

for i = 1:m

for j = 1:n

T1(i,j) = round(t(i,j)\*(F\_vrste(i)+F\_kolone(j))/2);

end

end

sumt\_vrste1 = [];

sumt\_vrste1 = sum(T1')

sumt\_kolone1 = sum(T1)

F\_vrste = T\_vrste./sumt\_vrste1

F\_kolone = T\_kolone./sumt\_kolone1

end

T1

F\_vrste

F\_kolone

% moguci razliciti rezultati zbog zaokruzivanja

**II resenje**

t=[0 20 40; 50 0 15; 40 15 0];

T\_vrste = [90 80 85];

T\_kolone = [120 55 75];

[m,n] = size(t);

sumt\_vrste = [];

sumt\_vrste = sum(t')

sumt\_kolone = sum(t)

F\_vrste = T\_vrste./sumt\_vrste

F\_kolone = T\_kolone./sumt\_kolone

T1 = [];

for i = 1:m

for j = 1:n

T1(i,j) = round(t(i,j)\*(F\_vrste(i)+F\_kolone(j))/2);

end

end

T1

sumt\_vrste1 = [];

sumt\_vrste1 = sum(T1')

sumt\_kolone1 = sum(T1)

F\_vrste1 = T\_vrste./sumt\_vrste1

F\_kolone1 = T\_kolone./sumt\_kolone1

for i = 1:m

for j = 1:n

T2(i,j) = round(T1(i,j)\*(F\_vrste1(i)+F\_kolone1(j))/2);

end

end

T2

sumt\_vrste2 = [];

sumt\_vrste2 = sum(T2')

sumt\_kolone2 = sum(T2)

F\_vrste2 = T\_vrste./sumt\_vrste2

F\_kolone2 = T\_kolone./sumt\_kolone2