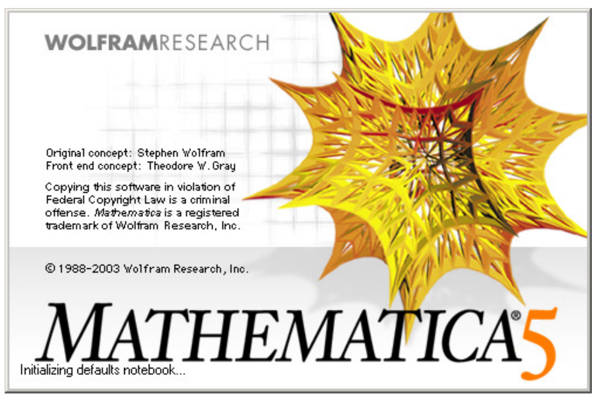
Универзитет Св. Кирил и Методиј – Р.Македонија

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО – СКОПЈЕ

Дискретна Математика 2
2011/2012



Лабораториски вежби:
Решенија на задачи
(во програмскиот пакет Mathematica 5.0)

Содржина

1. Лабораториски вежби 1: Вовед во Mathematica

Задача 1.1: Непарни броеви кои задоволуваат даден услов

Задача: Да се прикажат на екран сите непарни броеви i од 1 до n, за кои важи 3*i<n. n е број внесен од тастатура.

Решение:

```
n=Input["Vnesi go n"];
For[i=1,i<n/3,i++,If[Mod[i,2]==1,Print[i]]]
```

Задача 1.2: Делители на даден број

Задача: Внеси n броеви, и за секој од нив одреди ги неговите делители, и одреди го вкупниот број на делители. n е број внесен од тастатура.

Решение:

```
n=Input["n="];
brojDeliteli=0;
For[i=1,i<=n,i++,
  x=Input["broj="];
  Print["Deliteli na ",x, " se: "];
  For[j=2,j<=x/2,j++,If[Mod[x,j]==0,Print[j];brojDeliteli++;]]
]
Print["Vkupniot broj deliteli na ",n,"-te vneseni broevi e ",
         brojDeliteli]
```

Задача 1.3: Производ на цифри на даден број

Задача: Внеси n-цифрен број, каде n е број внесен од тастатура, и пресметај го производот на неговите цифри.

Решение:

```
n=Input["Broj na cifri: n="]
a=Input["Broj: a="]
rezultat=1;
tmp = a;
While[tmp>0,rezultat*=Mod[tmp,10];tmp=IntegerPart[tmp/10]];
Print["Proizvodot na cifri na ", n, "-cifreniot broj ", a, " e: ",
         rezultat];
```

2. Лабораториски вежби 2: Комбинаторика

Задача 2.1: Процедура за n!

Задача: Да се напише процедура за пресметување на n!

Решение (Кристијан Трајковски):

```
Faktoriel[n_]:=Module[{prod=1},
    For[i=1,i<=n,i++,prod*=i];
    prod
]
```

Задача 2.2: Процедура за биномен коефициент

Задача: Да се искористи процедурата од задача 1 за да се дефинира процедура која ќе го пресметува биномниот коефициент $\binom{n}{m}$.

Решение (Кристијан Трајковски): *(се користи процедирата Faktoriel од Задача 2.1)*

```
Binom[n_,k_]:=Module[{rezultat},
    rezultat=Faktoriel[n]/(Faktoriel[k]*Faktoriel[n-k]);
    rezultat
]
```

Задача 2.3: Процедура за Паскалов триаголник

Задача: Да се напише процедура за генерирање на Паскаловиот триаголник со n редици. Притоа треба да се искористи процедурата од задача 2. (*Помош*: Паскалов триаголник е триаголник во кој i-тата редица од триаголникот, i = 0, 1, …, n, ги содржи биномните коефициенти $\binom{i}{j}$ за j = 0, 1, …, i).

Решение 1: *(се користи процедирата Binom од Задача 2.2)*

```
Paskalov[n_]:=Module[{},
    For[i=0,i<=n,i++,
      red={};
      For[j=0,j<=i,j++,AppendTo[red,Binom[i,j]]];
      Print[red];
    ]
]
```

Решение 2 (Кристијан Трајковски): *(се користи процедирата Binom од Задача 2.2)*

```
Paskalov[n_]:=Module[{redStr=""},
    For[red=0, red<=n, red++,
      redStr="";
      For[broj=0, broj<=red, broj++,
```

```
        redStr = StringJoin[redStr, " ",ToString[Binom[red, broj]]];
        ];
      Print[redStr];
    ]
]
```

Задача 2.4: Процедура за пресметување број на пермутации без повторување

Задача: Да се напише процедура која ќе го пресметува бројот на пермутации без повторување од n елементи, класа k.

Решение (Кристијан Трајковски): *(се користи процедирата Faktoriel од Задача 2.1)*

```
PermBezPovt[n_, k_]:=Module[{rezultat},
    rezultat=Faktoriel[n]/(Faktoriel[n-k]);
    rezultat
]
```

Задача 2.5: Процедура за генерирање на пермутации без повторување

Задача: Да се напише процедура која ќе ги генерира сите можни пермутации без повторување од n елементи, класа k .

Решение 1:

```
<<DiscreteMath`Combinatorica`

SitePermBezPovtNew[n_, k_]:=Module[{arr={},site={}},
    For[i=1, i<=n, i++, AppendTo[arr, i]];
    arr=KSubsets[arr,k];
    For[i=1, i<Length[arr],i++,
        per=Permutations[arr[[i]]];
        For[m=1,m<=Length[per],m++,el=per[[m]];
          AppendTo[site,el]
         ];
    ];
    site
]
```

Решение 2 (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

SitePermBezPovt[n_, k_]:=Module[{arr={}, arr1},
    For[i=1, i<=n, i++, AppendTo[arr, i]];
    arr1=Subsets[arr];
    arr={};
    For[i=1, i<Length[arr1], i++,
      If[Length[arr1[[i]]]==k, AppendTo[arr, arr1[[i]]]]
```

```
    ];
    For[i=1, i<=Length[arr],i++,
      Print[Permutations[arr[[i]]]]
    ]
]
```

Задача 2.6: Процедура за пресметување број на комбинации без повторување

Задача: Да се напише процедура која ќе го пресметува бројот на комбинации без повторување од n елементи, класа k.

Решение: *(се користи процедирата Faktoriel од Задача 2.1)*

```
KombBezPovt[n_, k_]:=Module[{rezultat},
    rezultat=Faktoriel[n]/(Faktoriel[k]*Faktoriel[n-k]);
    rezultat
]
```

Задача 2.7: Процедура за генерирање на комбинации без повторување

Задача: Да се напише процедура која ќе ги генерира сите можни комбинации без повторување од n елементи, класа k .

Решение 1:

```
<<DiscreteMath`Combinatorica`

SiteKombBezPovt[n_, k_]:=Module[{arr={}},
    For[i=1, i<=n, i++, AppendTo[arr, i]];
    KSubsets[arr,k]
]
```

Решение 2 (Кристијан Трајковски):

```
SiteKombBezPovt[n_, k_]:=Module[{arr={}, arr1},
    For[i=1, i<=n, i++, AppendTo[arr, i]];
    arr1=Subsets[arr];
    arr={};
    For[i=1, i<Length[arr1], i++,
      If[Length[arr1[[i]]]==k, AppendTo[arr, arr1[[i]]]]
    ];
    arr
 ]
```

Задача 2.8: Процедура за пресметување број на пермутации со повторување

Задача: Да се напише процедура која ќе го пресметува бројот на пермутации со повторување од n елементи, класа k.

Решение:

```
PermSoPovt[n_, k_]:=Module[{rezultat},
    rezultat=n^k;
    rezultat
]
```

Задача 2.9: Процедура за генерирање на пермутации со повторување

Задача: Да се напише процедура која ќе ги генерира сите можни пермутации со повторување од n елементи, класа k

Решение:

```
<<DiscreteMath`Combinatorica`

SitePermSoPovt[n_,k_]:=Module[{t={},site={}},
    For[j=k,j<=n*k,j++,
        For[r=1,r<=n,r++,p=Partitions[j,r];
            For[i=1,i<=Length[p],i++,
                tmp=p[[i]];
                If[Length[tmp]==k,AppendTo[t,tmp]]
            ]
        ]
    ];
    t=Union[t];
    For[i=1,i<=Length[t],i++,
        per=Permutations[t[[i]]];
        For[m=1,m<=Length[per],m++,
            el=per[[m]];
            AppendTo[site,el]
        ];
    ];
    site
]
```

Задача 2.10: Процедура за пресметување број на комбинации со повторување

Задача: Да се напише процедура која ќе го пресметува бројот на комбинации со повторување од n елементи, класа k.

Решение: *(се користи процедирата Faktoriel од Задача 2.1)*

```
KombSoPovt[n_, k_]:=Module[{rezultat},
    rezultat=Faktoriel[n+k-1]/(Faktoriel[k]*Faktoriel[n-1]);
    rezultat
]
```

Задача 2.11: Процедура за генерирање на комбинации со повторување

Задача: Да се напише процедура која ќе ги генерира сите можни комбинации со повторување од n елементи, класа k

Решение: *(се користи процедурата SitePermSoPovt од Задача 2.9)*

```
<<DiscreteMath`Combinatorica`

SiteKombSoPovt[n_,k_]:=Module[{},
    per=SitePermSoPovt[n,k];
    Select[per,OrderedQ];
]
```

Задача 2.12: Процедура за број на пермутации без повторување за различни класи

Задача: Да се напише процедура која ќе го пресметува бројот на пермутации без повторување од n елементи, класа k, за секое k=1,2,…,m.

Решение: *(се користи процедирата PermBezPovt од Задача 2.4)*

```
PermBezPovtKlasi[n_, m_]:=Module[{},
    For[k=1,k<=m,k++,Print[PermBezPovt[n,k]]]
]
```

Задача 2.13: Процедура за број на комбинации без повторување за различни класи

Задача: Да се напише процедура која ќе го пресметува бројот на комбинации без повторување од n елементи, класа k, за секое k=1,2,…,m.

Решение: *(се користи процедирата KombBezPovt од Задача 2.6)*

```
KombBezPovtKlasi[n_, m_]:=Module[{},
    For[k=1,k<=m,k++,Print[KombBezPovt[n,k]]]
]
```

Задача 2.14: Процедура за број на пермутации со повторување за различни класи

Задача: Да се напише процедура која ќе го пресметува бројот на пермутации со повторување од n елементи, класа k, за секое k=1,2,…,m.

Решение: *(се користи процедирата PermSoPovt од Задача 2.8)*

```
PermSoPovtKlasi[n_, m_]:=Module[{},
    For[k=1,k<=m,k++,Print[PermSoPovt[n,k]]]
]
```

Задача 2.15: Процедура за број на комбинации со повторување за различни класи

Задача: Да се напише процедура која ќе го пресметува бројот на комбинации со повторување од n елементи, класа k, за секое k=1,2,…,m.

Решение: *(се користи процедирата KombSoPovt од Задача 2.10)*

```
KombSoPovtKlasi[n_, m_]:=Module[{},
    For[k=1,k<=m,k++,Print[KombSoPovt[n,k]]]
]
```

Задача 2.16: Процедура за генерирање пермутации со повторување од тип k1-k2-k3

Задача: Да се напише процедура која ќе ги генерира сите можни пермутации на елементите од множеството {a,b,c} во кои a се јавува k1 пати, b се јавува k2 пати, а c се јавува k3 пати.

Решение (Никола Соколов):

```
PermSoPovtTip1[k1_,k2_,k3_]:=Module[{l={}},
  For[i=1,i<=k1,i++,AppendTo[l,a]];
  For[i=1,i<=k2,i++,AppendTo[l,b]];
  For[i=1,i<=k3,i++,AppendTo[l,c]];
  Permutations[l]
]
```

Задача 2.17: Процедура за генерирање пермутации со повторување од тип k1-k2-k3-k4

Задача: Да се напише процедура која ќе ги генерира сите можни пермутации на елементите од множеството {1,2,3,4} во кои 1 се јавува k1 пати, 2 се јавува k2 пати, 3 се јавува k3 пати, и 4 се појавува k4 пати.

Решение (Никола Соколов):

```
PermSoPovtTip2[k1_,k2_,k3_,k4_]:=Module[{l={}},
  For[i=1,i<=k1,i++,AppendTo[l,1]];
  For[i=1,i<=k2,i++,AppendTo[l,2]];
  For[i=1,i<=k3,i++,AppendTo[l,3]];
  For[i=1,i<=k4,i++,AppendTo[l,4]];
  Permutations[l]
]
```

3. Лабораториски вежби 3: Релации

Задача 3.1: Проверка на својства на релации I

Задача: Нека R е релација на множество A={1,2,3,4,5,6}. Провери дали R е рефлексивна, антирефлексивна, симетрична, антисиметрична, транзитивна за следните релации:

a) a+b=5
b) a<b
c) a>=b
d) НЗД(a,b)=2
e) a-b>0
f) a+b е парен број
g) a*b е непарен број
h) Mod(a,b)=1
i) Mod(a,b)>3

Решение 1:

```
Proverka[r_,n_]:=Module[{},
    If[Tr[r]==n,Print["E refleksivna"],Print["Ne e refleksivna"]];
    If[Tr[r]==0,Print["E antirefleksivna"],Print["Ne e
antirefleksivna"]];
    If[Transpose[r]==r,Print["E simetricna"],Print["Ne e
simetricna"]];
    If[Sign[r+Transpose[r]-IdentityMatrix[n]]==r+Transpose[r]-
IdentityMatrix[n],Print["E antisimetricna"],Print["Ne e
antisimetricna"]];
    If[Sign[r+r.r]==r,Print["E tranzitivna"],Print["Ne e
tranzitivna"]];
]

A={1,2,3,4,5,6};
n=Length[A];

a=Table[If[i+j==5,1,0],{i,1,n},{j,1,n}];
MatrixForm[a]
Proverka[a,n]
```

```
b=Table[If[i<j,1,0],{i,1,n},{j,1,n}];
MatrixForm[b]
Proverka[b,n]

c=Table[If[i>=j,1,0],{i,1,n},{j,1,n}];
MatrixForm[c]
Proverka[c,n]

d=Table[If[GCD[i,j]==2,1,0],{i,1,n},{j,1,n}];
MatrixForm[d]
Proverka[d,n]

e=Table[If[i-j>0,1,0],{i,1,n},{j,1,n}];
MatrixForm[e]
Proverka[e,n]

f=Table[If[Mod[i+j,2]==0,1,0],{i,1,n},{j,1,n}];
MatrixForm[f]
Proverka[f,n]

g=Table[If[Mod[i*j,2]==1,1,0],{i,1,n},{j,1,n}];
MatrixForm[g]
Proverka[g,n]

h=Table[If[Mod[i,j]==1,1,0],{i,1,n},{j,1,n}];
MatrixForm[h]
Proverka[h,n]

i=Table[If[Mod[i,j]>3,1,0],{i,1,n},{j,1,n}];
MatrixForm[i]
Proverka[i,n]
```

Решение 2 (Михаил Петков):

```
a={};
For[i=1,i<=6,i++,red={};
  For[j=1,j<=6,j++,
    If[i+j==5,AppendTo[red,1],AppendTo[red,0]]];
  AppendTo[a,red]
]
MatrixForm[a]

If[Tr[a]==6, Print["Relacijata a e Refleksivna"],
   Print["Relacijata a ne e Refleksivna"]]

If[Tr[a]==0, Print["Relacijata a e Antirefleksivna"],
   Print["Relacijata a ne e Antirefleksivna"]]

If[Transpose[a]==a, Print["Relacijata a e Simetricna"],
   Print["Relacijata a ne e Simetricna"]]

If[Sign[a+Transpose[a]-IdentityMatrix[6]]==
  a+Transpose[a]-IdentityMatrix[6],
  Print["Relacijata a e Antisimetricna"],
  Print["Relacijata a ne e Antisimetricna"]]

If[Sign[a+a.a]==a,Print["Relacijata a e Tranzitivna"],
  Print["Relacijata a ne e Tranzitivna"]]
```

Задача 3.2: Проверка на својства на релации II

Задача: Нека R е релација на множество A={1,2,3,4}, провери дали R е рефлексивна, антирефлексивна, симетрична, антисиметрична, транзитивна за следниве релации:

a) {(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)}
b) {(1, 1), (1 , 2), (2, 1), (2, 2), (3, 3), (4, 4)}
c) {(2, 4), (4, 2)}
d) {(1, 2), (2, 3), (3, 4)}
e) {(1, 1), (2, 2), (3, 3), (4, 4)}
f) {(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)}
g) {(1,1), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (4,4)}

Последната релација проверете ја и за множеството A={1,2,3,4,5}

Решение: *(се користи процедурата Proverka од Задача 3.1)*

```
A={1,2,3,4};
n=Length[A];

a=Table[0,{i,1,n},{j,1,n}];
r={{2,2},{2,3},{2,4},{3,2},{3,3},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];a[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[a]
Proverka[a,n]

b=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{1,2},{2,1},{2,2},{3,3},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];b[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[b]
Proverka[b,n]

c=Table[0,{i,1,n},{j,1,n}];
r={{2,4},{4,2}}
For[i=0,i<Length[r],i++;tmp=r[[i]];c[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[c]
Proverka[c,n]

d=Table[0,{i,1,n},{j,1,n}];
r={{1,2},{2,3},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];d[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[d]
Proverka[d,n]

e=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{2,2},{3,3},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];e[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[e]
Proverka[e,n]

f=Table[0,{i,1,n},{j,1,n}];
r={{1,3},{1,4},{2,3},{2,4},{3,1},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];f[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[f]
Proverka[f,n]

g=Table[0,{i,1,n},{j,1,n}];
```

```
r={{1,1},{2,2},{2,3},{2,4},{3,2},{3,3},{3,4},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];g[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[g]
Proverka[g,n]

A1={1,2,3,4,5};
n1=Length[A1];
g1=Table[0,{i,1,n1},{j,1,n1}];
r={{1,1},{2,2},{2,3},{2,4},{3,2},{3,3},{3,4},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];g1[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[g1]
Proverka[g1,n]
```

Задача 3.3: Проверка за релација за подредување

Задача: Нека R е релација на множество A={1,2,3,4}, провери дали R е релација за подредување:

a) {(1,1), (1,2), (2,1), (2,2), (3,3), (4,4)}
b) {(1,2), (2,3), (3,4)}
c) {(1,1), (2,2), (3,3), (4,4)}
d) {(1,3), (1,4), (2,3), (2,4), (3,1), (3,4)}
e) {(1,1), (2,2), (2,3), (2,4), (3,3), (3,4), (4,4)}

Решение 1:

```
Podreduvanje[r_,n_]:=Module[{},
   If[Tr[r]==n && Sign[r+Transpose[r]-
IdentityMatrix[n]]==r+Transpose[r]-IdentityMatrix[n] &&
Sign[r+r.r]==r,
      Print["E relacija za podreduvanje"],Print["Ne e relacija za
podreduvanje"];
   ]
]

A={1,2,3,4};
n=Length[A];

a=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{1,2},{2,1},{2,2},{3,3},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];a[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[a]
Podreduvanje[a,n]

b=Table[0,{i,1,n},{j,1,n}];
r={{1,2},{2,3},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];b[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[b]
Podreduvanje[b,n]

c=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{2,2},{3,3},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];c[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[c]
Podreduvanje[c,n]
```

```
d=Table[0,{i,1,n},{j,1,n}];
r={{1,3},{1,4},{2,3},{2,4},{3,1},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];d[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[d]
Podreduvanje[d,n]

e=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{2,2},{2,3},{2,4},{3,3},{3,4},{4,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];e[[tmp[[1]],tmp[[2]]]]=1]
MatrixForm[e]
Podreduvanje[e,n]
```

Решение 2 (Суад Саљиу и Басри Јашари):

```
m={{0,1,0,0},{0,0,1,0},{0,1,1,1},{0,0,0,1}}
MatrixForm[m]
If[Tr[m]==4 && Sign[m+Transpose[m]-IdentityMatrix[4]]==
  m+Transpose[m]-IdentityMatrix[4] && Sign[m+m.m]==m,
  Print["E relacija za podreduvanje"],
  Print["Ne e relacija za podreduvanje"]
]
```

Задача 3.4: Наоѓање на транзитивно проширување (алгоритам на Варшал)

Задача: Најди го транзитивното проширување на следните релации од множеството {1,2,3,4} според алгоритмот на Варшал:

a) {(1,2),(2,1),(2,3),(3,4),(4,1)}
b) {(2,1),(2,3),(3,1),(3,4),(4,1),(4,3)}
c) {(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)}
d) {(1,1),(1,4),(2,1),(2,3),(3,1),(3,2),(3,4),(4,2)}

Решение 1:

```
Warshall[r_,n_]:=Module[{x=r},
  Print["W0=",MatrixForm[x]];
  For[k=1,k<=n,k++,
    For[i=1,i<=n,i++,
      If[x[[k,i]]==1,
        For[j=1,j<=n,j++,If[x[[j,k]]==1,x[[j,i]]=1]]
      ]
    ];
    Print["W",k,"=",MatrixForm[x]]
  ]
]

A={1,2,3,4};
n=Length[A];

a=Table[0,{i,1,n},{j,1,n}];
r={{1,2},{2,1},{2,3},{3,4},{4,1}}
For[i=0,i<Length[r],i++;tmp=r[[i]];a[[tmp[[1]], tmp[[2]]]]=1]
Warshall[a,n]
```

```
b=Table[0,{i,1,n},{j,1,n}];
r={{2,1},{2,3},{3,1},{3,4},{4,1},{4,3}}
For[i=0,i<Length[r],i++;tmp=r[[i]];b[[tmp[[1]], tmp[[2]]]]=1]
Warshall[b,n]

c=Table[0,{i,1,n},{j,1,n}];
r={{1,2},{1,3},{1,4},{2,3},{2,4},{3,4}}
For[i=0,i<Length[r],i++;tmp=r[[i]];c[[tmp[[1]], tmp[[2]]]]=1]
Warshall[c,n]

d=Table[0,{i,1,n},{j,1,n}];
r={{1,1},{1,4},{2,1},{2,3},{3,1},{3,2},{3,4},{4,2}}
For[i=0,i<Length[r],i++;tmp=r[[i]];d[[tmp[[1]], tmp[[2]]]]=1]
Warshall[d,n]
```

Решение 2 (Суад Саљиу и Басри Јашари):

```
a={{0,1,0,0},{1,0,1,0},{0,0,0,1},{1,0,0,0}}
For[i=1,i≤4,i++,
  For[m=1,m≤4,m++,
    If[a[[m,i]]==1,
      For[j=1,j≤4,j++,
        If[a[[j,m]]==1,a[[j,i]]=1]
      ]
    ]
  ]
]
MatrixForm[a]
```

Задача 3.5: Наоѓање на транзитивно проширување

Задача: Најди го транзитивното проширување на следниве релации од множеството {a,b,c,d,e}:

a) {(a , c), (b, d), (c, a), (d, b), (e, d)}
b) {(b, c), (b, e), (c, e), (d, a), (e, b), (e, c)}
c) {(a , b), (a , c), (a , e), (b, a), (b, c), (c, a), (c, b), (d, a),(e, d)}
d) {(a , e), (b, a), (b, d), (c, d), (d, a), (d, c), (e, a), (e, b),(e, c), (e, e)}

Решение : *(се користи процедурата Warshall од Задача 3.4)*

```
A={a,b,c,d,e};
n=Length[A];

Print["R={(a,c),(b,d),(c,a),(d,b),(e,d)}"];
a={{0,0,1,0,0},{0,0,0,1,0},{1,0,0,0,0},{0,1,0,0,0},{0,0,0,0,1}};
Warshall[a,n]
```

4. Лабораториски вежби 4: Графови

Задача 4.1: Визуелно претставување на графови

Задача: Да се претстават визуелно графовите G = (V, E) каде темињата i и j (означени со броеви) (i, j $\in$ V), |V|=n, од графот G се поврзани со ребро ако :

a) n=7, остатокот при делење на бројот i со бројот j е различен од 1
b) n=10, остатокот при делење на бројот j со бројот i е помал од 3
c) n=12, целиот дел од резултатот при делење на бројот i со бројот j е помал од 3 или поголем од 5
d) n=20, збирот од квадратот на бројот i и бројот i е поголем од разликата на кубот на бројот j и бројот j
e) n=15, разликата на бројот i и целиот дел од неговиот квадратен корен е поголема и еднаква од остатокот при делење на бројот j со |j-i|

Решение (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

n=7
a=MakeGraph[Range[n],(Mod[#1,#2]==1)&]
ShowGraph[a]

n=10
b=MakeGraph[Range[n],(Mod[#1,#2]<3)&]
ShowGraph[b]

n=12
c=MakeGraph[Range[n],(Floor[#1/#2]>5||Floor[#1/#2]<3)&]
ShowGraph[c]

n=20
d=MakeGraph[Range[n],(#1^2+#1>#2^3-#2)&]
ShowGraph[d]

n=15
e=MakeGraph[Range[2,n+1],(#1-IntegerPart[Sqrt[#1]]>=Mod[#2,(#2-
1)])&]
ShowGraph[e]
```

Задача 4.2: Комплетен граф, граф ѕвезда, опции за графови, додавање теме, бришење ребро

Задача: Нацртајте комплетен граф од ред n (n>3) и граф ѕвезда од ред m (m>4), притоа првите две темиња од графовите нека бидат со сина боја, последното теме со црвена боја, а ребрата со зелена боја. Првите 2 темиња нека бидат визуелно поголеми. Потоа кај комплетниот граф додадете едно теме помеѓу i-тото и j-тото теме, а кај графот ѕвезда избришете го реброто помеѓу овие две темиња. (n, m, i и j се внесуваат преку тастатура).

Решение 1 (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

n = Input["Vnesi N"]
m = Input["Vnesi M"]
i = Input["Vnesi I"]
j = Input["Vnesi J"]
completegraph = CompleteGraph[n]
ShowGraph[SetGraphOptions[AddVertex[completegraph, {i, j}],
  {{1, 2, VertexColor->Blue, VertexStyle->Disk[Large]},
  {n, VertexColor-> Red}}, EdgeColor->Green]
]
stargraph=Star[m]
ShowGraph[DeleteEdge[stargraph, {i, j}]]
```

Решение 2 (Ведрана Тозија):

```
<<DiscreteMath`Combinatorica`

n=Input["Vnesete broj na teminja(pogolem od 3) za kompletniot graf"]
g1=SetGraphOptions[CompleteGraph[n],
   {{1,2,VertexColor->Blue,VertexStyle->Disk[Large]},
    {n,VertexColor->Red}},
   EdgeColor->Green
 ];
ShowGraph[g1]

m=Input["Vnesete broj na teminja (pogolem od 4) za grafot zvezda"]
g2=SetGraphOptions[Star[m],
   {{1,2,VertexColor->Blue,VertexStyle->Disk[Large]},
    {m,VertexColor->Red}},
   EdgeColor->Green, VertexNumber->True
 ];
ShowGraph[g2]

i=Input["Prvo teme"]
j=Input["Vtoro teme"]

a=AddVertex[g1,{i,j}];
ShowGraph[a]

b=DeleteEdge[g2,{i,j}];
ShowGraph[b]
```

Задача 4.3: Графови со алки, унија на графови, подграф од даден ранг

Задача: Нацртајте 2 графа со алки (лупи) со помош на наредбата MakeGraph, а потоа определете ја позицијата на темињата и бројот на ребра. Додадете n rebra и m темиња кај првиот граф и избришете m ребра и n темиња кај вториот граф. Потоа најдете унија на двата графа, и од новодобиениот граф претставете подграф од ранг t (n, m и t се броеви внесени од тастатура).

Решение (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

n = Input["Vnesi N"]
m = Input["Vnesi M"]
t = Input["Vnesi T"]

loopgraph1 = MakeGraph[Range[n], (#1==#2)&]
loopgraph2 = MakeGraph[Range[m], (#1==#2)&]
For[i=1, i<n, i++,
  loopgraph1 = AddEdge[loopgraph1, {i, i+1}];
  loopgraph2 = DeleteVertex[loopgraph2, i];
]
For[i=0, i<m, i++,
  loopgraph1 = AddVertex[loopgraph1];
  loopgraph2 = DeleteEdge[loopgraph2, {i, i}];
]
ShowGraph[loopgraph1]
ShowGraph[loopgraph2]

uniongraph=GraphUnion[loopgraph1, loopgraph2]
ShowGraph[uniongraph]

ShowGraph[InduceSubgraph[uniongraph, RandomSubset[Range[t]]]]
```

Задача 4.4: Цртање граф преку мени за избор

Задача: Овозможете корисникот преку мени за избор да нацрта комплетен граф или ѕвезда граф од произволен ред и потоа да има можност да избере додавање, бришење на произволен број на темиња и ребра. И во зависниот од изборот да се нацрта конечниот граф.

Решение (Кристијан Трајковски):

```
While[1==1,
  opcija=Input["Izberi Opcija"];
  If[opcija==0, Break[]];
  If[opcija==1,
    n=Input["vnesi n"];
    graph=CompleteGraph[n];
  ];
  If[opcija==2,
      n=Input["vnesi n"];
      graph=Star[n];
  ];
  If[opcija==3,
    graph=AddVertex[graph]
  ];
  If[opcija==4,
    index = Input["Vnesi indeks"];
    graph=DeleteVertex[graph, index]
  ];
  If[opcija==5,
```

```
    teme1 = Input["Vnesi prvo teme"];
    teme2 = Input["Vnesi vtoro teme"];
    graph=AddEdge[opcija, {teme1, teme2}]
  ];
  If[opcija==6,
    teme1 = Input["Vnesi prvo teme"];
    teme2 = Input["Vnesi vtoro teme"];
    graph=DeleteEdge[graph, {teme1, teme2}]
  ]
]
ShowGraph[graph]
```

Задача 4.5: Проверка на изоморфизам на графови

Задача: Проверете изоморфизам кај:

a) Два произволни графа генерирани преку наредбата MakeGraph
b) Два комплетни графа со произволен број на елементи во двете дисјунктни множества на темиња во двата графа поединечно.

Решение (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

graf1=MakeGraph[Range[5],(#1==#2 || #1<#2)&];
graf2=MakeGraph[Range[5],(#1==#2 || #1>#2)&];
Isomorphism[graf1,graf2,All]

graf3=CompleteGraph[5]
graf4=CompleteGraph[5]
Isomorphism[graf3,graf4,All]
```

Задача 4.6: Матрица на соседство за комплетен граф, степени на темиња, број на ребра

Задача: Креирајте матрица на соседство за комплетен граф од ред n (темињата означете ги со броеви), а потоа преку користење на матрицата најдете ја сумата на степени од сите темиња, а потоа искористете го тој резултат за да го добиете бројот на ребра.

Решение (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

n=Input["Vnesi vrednost za n:"]
graf=CompleteGraph[n]
table=TableForm[ToAdjacencyMatrix[graf]]
Vertices[graf]

brojac=0;
suma=0;
For[i=1,i<=n,i++,
```

```
  brojac=0;
  For[j=1, j<=n, j++,
    If[table[[1, i, j]]==1 || table[[1,j,i]]==1,brojac++];
  ];
  suma+=brojac;
]
Print[suma]
rebra=suma/2
Print[rebra]
```

Задача 4.7: Матрица на соседство за произволен граф, степени на темиња, број на ребра

Задача: Креирајте матрица на соседство за произволен ориентиран граф користејќи ја наредбата MakeGraph (темињата означете ги со броеви), а потоа преку користење на матрицата најдете ја сумата на излезни и влезни степени од сите темиња, а потоа искористете го тој резултат за да го добиете бројот на ребра.

Решение (Кристијан Трајковски):

```
<<DiscreteMath`Combinatorica`

n=5
(*graph=MakeGraph[Range[4], (True)&]*)
graph=OrientGraph[RandomGraph[n, 0.85]]
ShowGraph[graph]
tabela=TableForm[ToAdjacencyMatrix[graph]]
izleg=0
vleg=0
For[i=1, i<=n, i++,
  For[j=1, j<=n, j++,
    If[tabela[[1,i,j]]==1, izleg++];
    If[tabela[[1,j,i]]==1, vleg++];
  ]
]
rebra=izleg
Print[rebra]
```

5. Лабораториски вежби 5: Булова алгебра

Задача 5.1: Вредности на Булови функции

Задача: Најдете ги вредностите на следните Булови функции:

a) $F(x,y) = xy + \neg(\neg x + \neg xy)$
b) $F(x,y) = x + y + (\neg xy) \oplus (x+y)$
c) $F(x,y) = xy + (x \oplus y)$
d) $F(x,y) = x\,(y\neg x + yx)\;1\;(y\neg x + yx) + 0$
e) $F(x,y) = \neg x\,(y + yx) + 0x$

Решение:

(Виктор Петровски)

```
a[x_,y_]:= (x∧y)∨(¬(¬x∨¬x∧y))
a[True,True]
a[True,False]
a[False,True]
a[False,False]
```

```
b[x_,y_]:=x ∨y ∨(¬x∧y)⊻(x∨y)
b[True,True]
b[True,False]
b[False,True]
b[False,False]
```

(Горјан Јовановски и Бојан Јаневски)

```
c[x_,y_]:=x&&y||Xor[x,y]
c[True,True]
c[True,False]
c[False,True]
c[False,False]
```

```
d[x_,y_]:=x&&(y&&!x||y&&x)&&True&&(y&&!x||y&&x)||False
d[True,True]
d[True,False]
d[False,True]
d[False,False]
```

```
e[x_,y_]:= ¬x ∧(y ∨ y∧x)∨False∧x
e[True,True]
e[True,False]
e[False,True]
e[False,False]
```

Задача 5.2: Еквиваленција помеѓу Булови изрази

Задача: Проверете дали важат еквиваленциите помеѓу овие изрази:

a) $x \oplus y = (x + y)\ \neg(xy)$
b) $x \oplus y = (x\neg y) + (\neg xy)$
c) $x \oplus y = (x + \neg y)\ \neg\ (x\neg y)$

Решение (Ненад Стојаноски и Трајче Петрески):

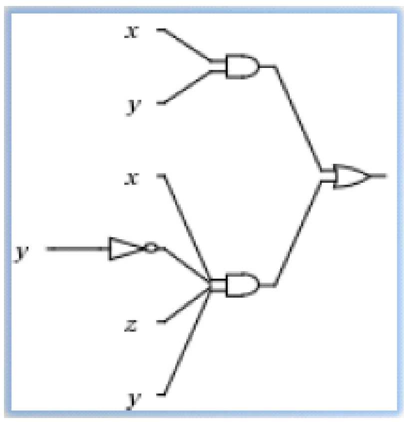
```
Ekvivalencija[f_,g_]:=Module[{},
   If[f[True,True]==g[True,True] &&
      f[True,False]==g[True,False] &&
      f[False,True]==g[False,True] &&
      f[False,False]==g[False,False],
        Print["Izrazite se ekvivalentni"],
        Print["Izrazite ne se ekvivalentni"]
   ]
]
```

```
IsklucivoIli[x_,y_]:=(x&&!y)||(!x&&y)
f1[x_,y_]:=(x||y)&&(!(x&&y))
f2[x_,y_]:=(x&&!y)||(!x&&y)
f3[x_,y_]:=(x||!y)&&(!(x&&!y))

Ekvivalencija[IsklucivoIli,f1]
Ekvivalencija[IsklucivoIli,f2]
Ekvivalencija[IsklucivoIli,f3]
```

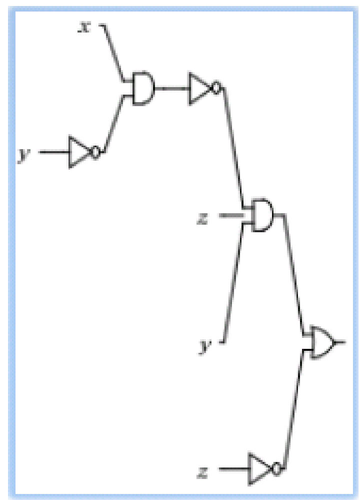
Задача 5.3: Логички изрази и логички кола

Задача: Дефинирајте ги логичките изрази со логички оператори во Mathematica за следните логички кола:

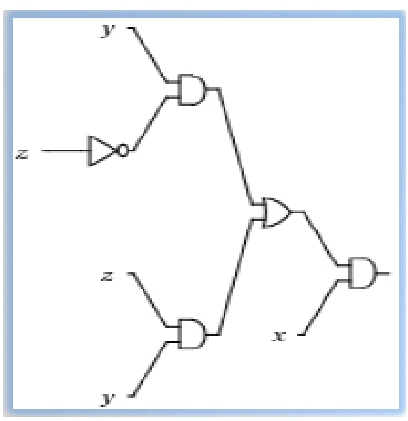
a)



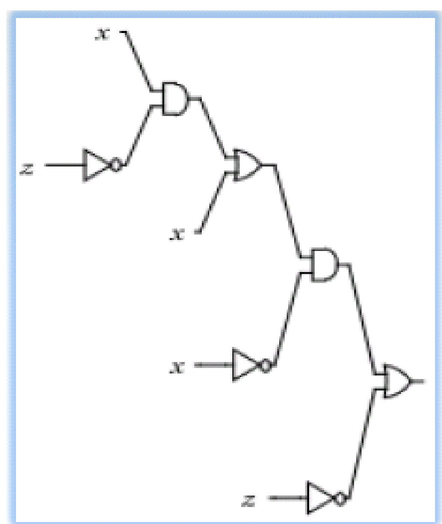
b)



c)



d)



Решение (Виктор Петровски):

```
a[x_,y_,z_]=(x∧y) ∨(x ∧¬y∧z∧y)
b[x_,y_,z_]=(¬(x∧¬y)∧(z∧y)) ∨(¬z)
c[x_,y_,z_]=((y∧¬z) ∨(z∧y))∧ x
d[x_,y_,z_]=(((x∧¬z) ∨ x)∧¬x) ∨¬z
```

Задача 5.4: Логички порти и логички кола

Задача: За следниве изрази, испрограмирајте функции кои ќе работат како логички кола, односно ќе примаат соодветен број на нули или единици. Пожелно е да направите ваши функции за базните порти (НЕ/ИЛИ/И).

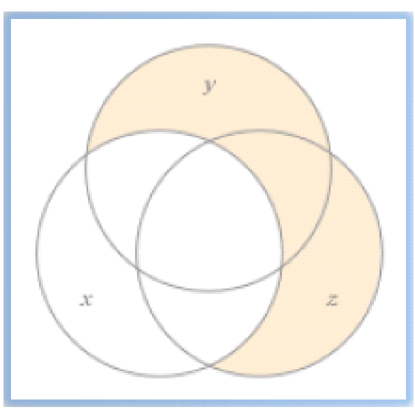
Решение (Никола Соколов):

```
IPorta[x_,y_]:=Module[{res},If[x==1&&y==1,res=1,res=0];res]
ILIPorta[x_,y_]:=Module[{res},If[x==1||y==1,res=1,res=0];res]
NEPorta[x_]:=Module[{res},If[x==0,res=1,res=0];res]

Kolo1[x_,y_]:=ILIPorta[IPorta[x,NEPorta[y]],IPorta[NEPorta[x],y]]

Kolo2[x_,y_]:=NEPorta[IPorta[x,y]]

Kolo3[x_,y_]:=NEPorta[ILIPorta[x,y]]

Kolo4[x_,y_]:=ILIPorta[IPorta[x,y],NEPorta[ILIPorta[x,IPorta[x,y]]]]

Kolo5[x_,y_,z_]:=ILIPorta[IPorta[NEPorta[IPorta[NEPorta[x],y]],
  ILIPorta[x,z]],IPorta[IPorta[NEPorta[x],y],
  NEPorta[ILIPorta[x,z]]]]

Kolo6[x_,y_]:=ILIPorta[IPorta[x,y],ILIPorta[IPorta[NEPorta[x],y],
  IPorta[x,NEPorta[y]]]]

Kolo7[x_,y_]:=ILIPorta[IPorta[x,ILIPorta[IPorta[y,NEPorta[x]],
  IPorta[y,x]]],ILIPorta[IPorta[y,NEPorta[x]],IPorta[y,x]]]

Kolo8[x_,y_]:=NEPorta[ILIPorta[IPorta[NEPorta[x],ILIPorta[y,
  IPorta[y,x]]],y]]
```

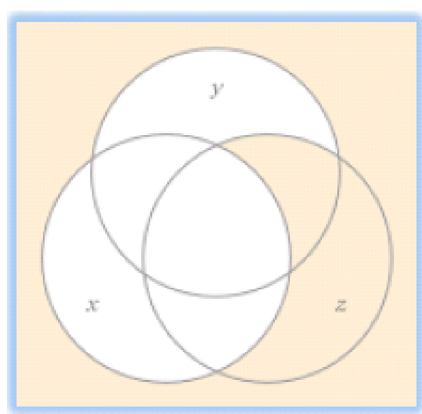
Задача 5.5: Булови изрази и Венови дијаграми

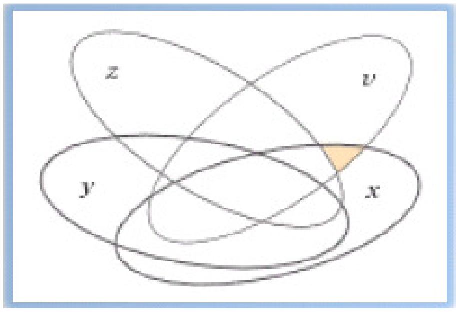
Задача: Потрудете се да најдете Булови изрази кои ги задоволуваат следниве Венови дијаграми. Дефинирајте ги изразите и тестирајте ги за одредени вредности. Внимавајте на обоените позадини.

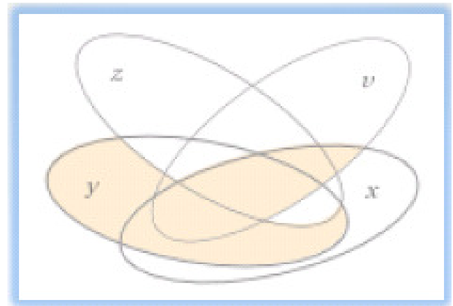
a)



b)



c) 

d) 

Решение (Горјан Јовановски и Бојан Јаневски):

```
a[x_,y_,z_]:=(y||z)&&!x
b[x_,y_,z_]:=(z&&!x)||!(x||y)
c[x_,y_,z_,v_]:=x&&v&&!z&&!y
d[x_,y_,z_,v_]:=(y&&!z)||(x&&v)
```