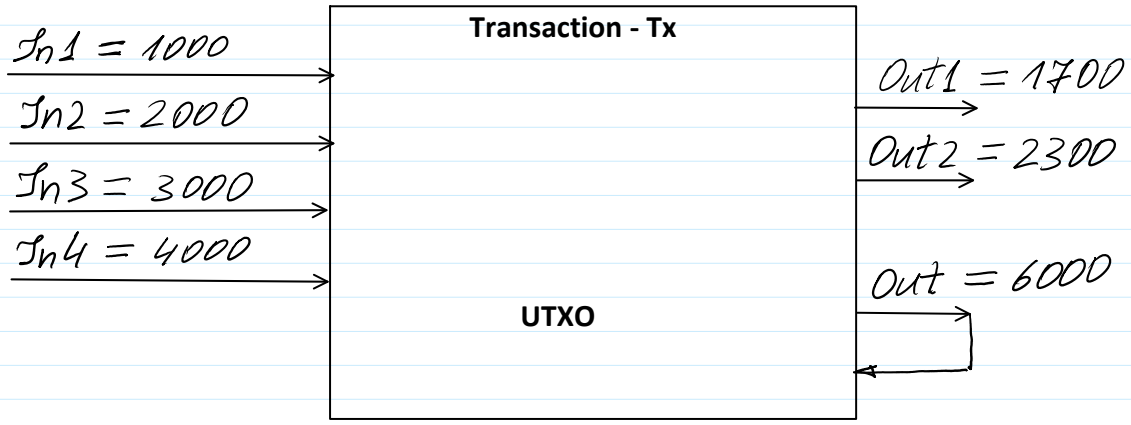
Interactive Exam Tasks:
1.Transaction creation according to template.
2.Public and private and public keys generation using **PP**=(**p**, **g**); **p**=264043379, **g**=2.
3.Transaction signing using Schnorr signature.
4.Arbitrary 3 transactions including in the Merkle Tree.
5.To create a block and mine-validate created bloc.

Buhalterija - Book-Keeping, Accounting
Unspend Transaction Output - UTXO

No.	Pajamos-Incomes	Išlaidos-Expenses	Likutis-Balance
In1.	Client1: 1000 Sat		1000 Sat
In2.	Client2: 2000 Sat	Out1. Firm 5: 1700 Sat	1300 Sat
In3.	Client3: 3000 Sat	Out2. Firm 6: 2300 Sat	2000 Sat
In4.	Client4: 4000 Sat	Out3. Firm 7:	6000 Sat
Total	**10 000 Sat**	**4000 Sat**	**6000 Sat**

Sum of Inputs =
= Sum of Outputs
Divisibility



Public Parameters **PP = (p, g);** **p**=264043379; **g**=2;

A : PrK_A=x; PuK_A=a;

>> x=randi(2^27)
x = 100497451
>> a=mod_exp(g,x,p)
a = 91968695

B: PrK_B=y; PuK_B=b;

>> y=randi(2^27)
y = 25824381
>> b=mod_exp(g,y,p)
b = 195335035

Schnorr Signature ← M - message to be signed
Signature is denoted by **Ϭ=S=(r, s)**

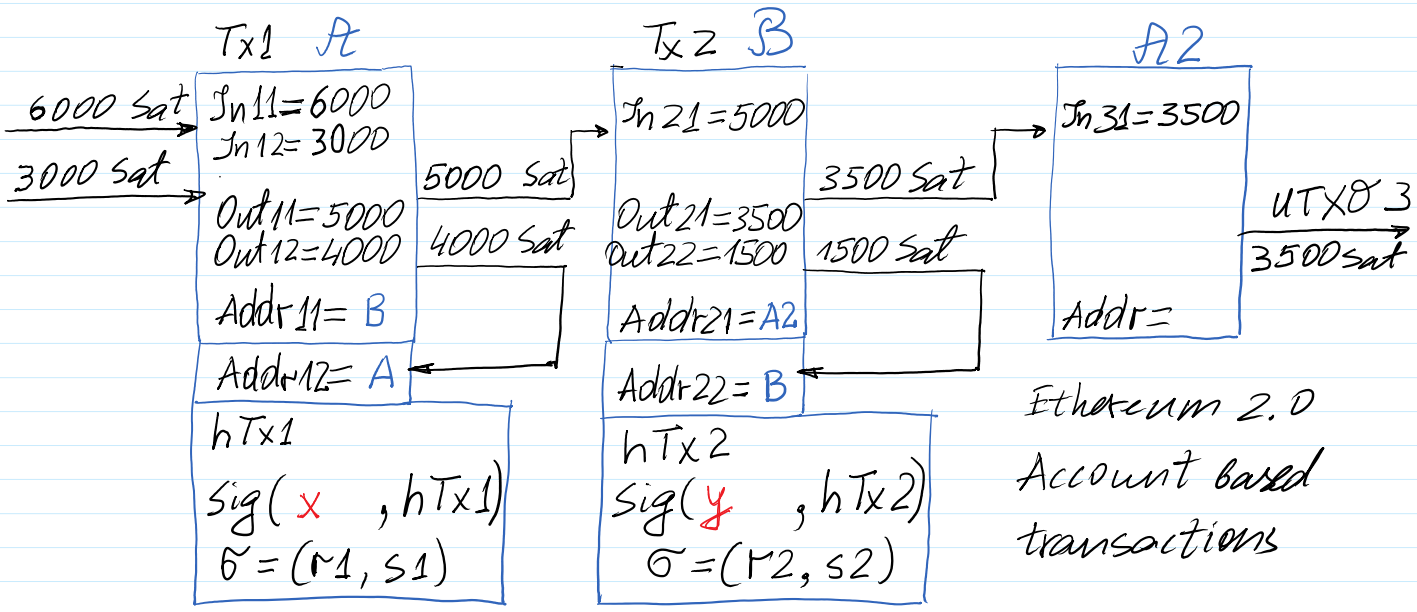
u=randi(p-1)

Verification of **Ϭ=S=(r,s)** for **h=H(M||r)**

u=randi(p-1)
r=g^u mod p
h=H(M||r) **%** **h** is a decimal number
s=u+xh mod (p-1)

Verification of **σ=S=(r,s)** for **h=H(M||r)**

$$g^s = r a^h \bmod p$$



'Tx1 : In11 = 6000 || In12 = 3000 || Out11 = 5000 || Out12 = 4000 || Rec1=A2 || Rec2=B'

$hTx1 = h28(\downarrow)$

$Sign(PrK_A = x, hTx1) =$ $S_1 = \sigma_1 = (r_1, s_1)$ % Schnorr Sig.

'Tx2: In21=5000 || Out21=3500 || Out22=1500 || Rec1=A2 || Rec2=B'

$hTx2 = h28('\downarrow')$

$sign(PrK_B = y, h2Sig) =$ $S_2 = \sigma_2 = (r_2, s_2)$ % Schnorr Sig.

Transaction templates:
Tx1 = 'Tx1:In11=6000||In12=3000||Out11=5000||Out12=4000||Rec1=B||Rec2=A'

Tx2='Tx2:In21=5000||Out21=3500||Out22=1500||Rec1=B2||Rec2=B'

```
>> hTx1=h28('Tx1:In11=6000||In12=3000||Out11=5000||Out12=4000||Rec1=B||Rec2=A')
hTx1 = AFC73D8
>> hTx1=h28(Tx1)
hTx1 = AFC73D8
>> hTx1d=hd28(Tx1)
```

hTx1d = 184316888

```
>> hex2dec('AFC73D8')
ans =  184316888
>> hex2dec(hTx1)
ans =  184316888
```

```
>>  hTx2=h28('Tx2:In21=5000||Out21=3500||Out22=1500||Rec1=B2||Rec2=B')
hTx2 = 43BC2D0
>> hTx2=h28(Tx2)
hTx2 = 43BC2D0
>> hTx2d=hd28(Tx2)
hTx2d =  71025360
>> hex2dec(hTx2)
ans =  71025360
```

Schnorr Signature ← M - message to be signed in decimal format

Transaction Tx1 signing: **M=hTx1d =** 184316888

hTx1d=hd28(Tx1)
u1=randi(p-1)
r1=mod_exp(g,u1,p)
h1=hd28(hTx1d||r1) **%** **h** is a decimal number
s1=u1+x(h1) mod (p-1)

Signature fot Tx1 computation σ_1=**S1**=(**r1**, **s1**) Signature fot Tx2 computation σ_2=**S2**=(**r2**, **s2**)

```
>> u1=randi(p-1)
u1 =  96927099
>> r1=mod_exp(g,u1,p)
r1 = 240239134
>> h1=hd28(hTx1d||r1)
h1 =  126174618
>> xh1=mod(x*h1,p-1)
xh1 =  319146
>> s1=mod(u1+xh1,p-1)
s1 =  97246245
```

$$g^s = ra^h \bmod p$$

$$g^{s1} = (r1)a^{h1} \bmod p$$

```
>> g_s1=mod_exp(g,s1,p)
g_s1 = 168542564
>> a
a = 91968695
```

```
>> a_h1=mod_exp(a,h1,p)
a_h1 = 128485502
>> r1
r1 = 240239134
>> r1a_h1=mod(r1*a_h1,p)
r1a_h1 = 168542564
```

```
>> hTx1d=hex2dec('AFC73D8')
hTx1d =  184316888
```

%%% Signature **Sig1** Creation on **hTx1d**
Transaction Tx1 signing: **M=hTx1d**

```
u1=randi(p-1)
r1=mod_exp(g,u1,p)
h1=hd28('hTx1d||r')    % h1 is a decimal number
xh1=mod(x*h1,p-1)
s1=mod(u1+xh1,p-1)
```

Sig1=(r1, s1)

Signature **Sig1** Verification

%%% Signature verification

```
>> hTx2d=hex2dec('43BC2D0')
hTx2d =  71025360
```

%%% Signature **Sig2** Creation on **hTx2d**
Transaction Tx2 signing: **M=hTx2d**

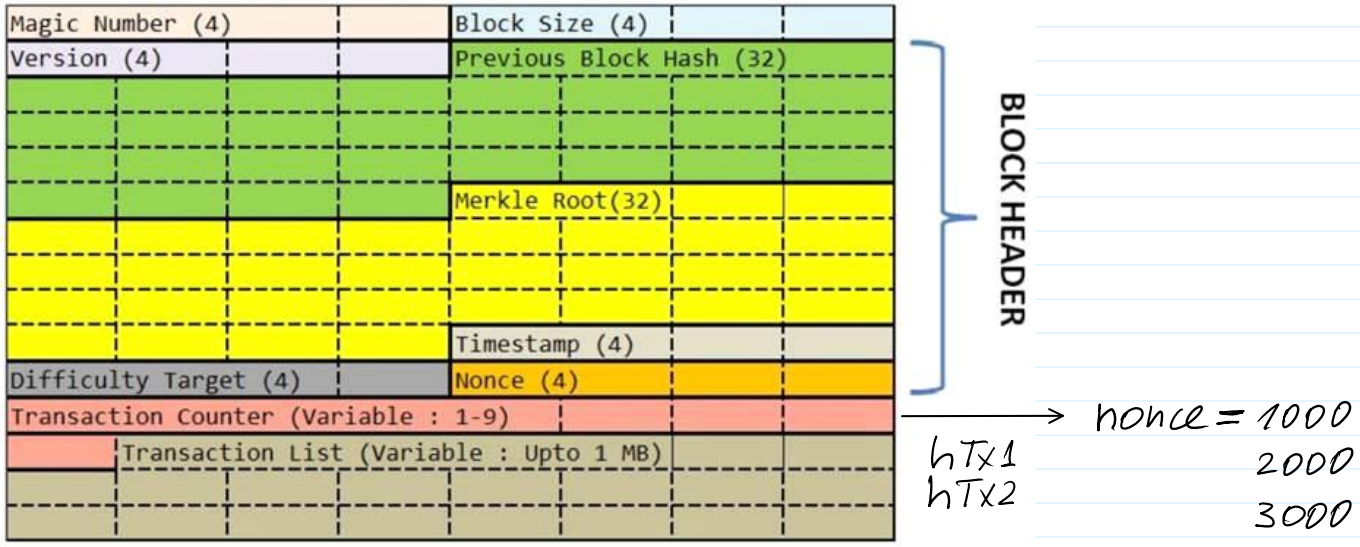
```
u2=randi(p-1)
r2=mod_exp(g,u1,p)
h2=hd28('hTx2d||r')    % h1 is a decimal number
xh2=mod(y*h2,p-1)
s1=mod(u1+xh2,p-1)
```

Sig2=(r2, s2)

Signature **Sig2** Verification

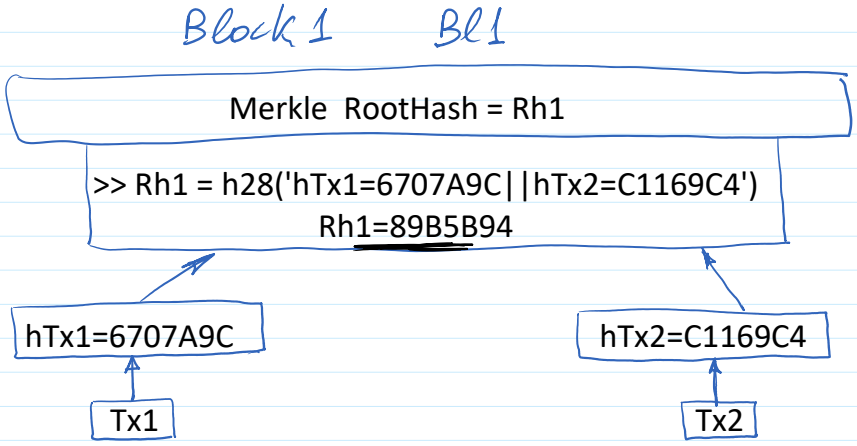
%%% Signature verification

…… Till this place ……




```
>> PrBlh=h28('nonce=1020')
```

PrBlh = 0CAF06F



Dificulty target DT = ϕ HHHHHH

Block 1: PrBlh: PrBlh=0CAF06F

Rh : Rh=89B5B94

hTx1=6707A9C

hTx2=C1169C4

nonce : nonce=1000
nonce=2000
nonce=3000
.....

'Bl1 : PrBl = ... || Rh1 = ... || hTx1 = ... || hTx2 = ... || nonce = 1000'

```
>> Bl1M=h28('Bl1:PrBlh=0CAF06F||Rh=89B5B94||hTx1=6707A9C||hTx2=C1169C4||nonce=1000')
Bl1M = D99C1E7

>> Bl1M=h28('Bl1:PrBlh=0CAF06F||Rh=89B5B94||hTx1=6707A9C||hTx2=C1169C4||nonce=2018')
Bl1M = 06B0772
```

h28 : computes h - value of 28 bits

According to the difficulty target the adequate mined block must have leading hexadecimal 0 or 4 leading 0_b bits.

All available h - values with 28 bits length is equal to

```
>> 2^28   ans =  268435456
```

The number of adequate mined values is equal to any 28 - 4 = 24 bits values

```
>> 2^24   ans =  16777216
```

The probability to mine a block is the following:

$$Pr = \frac{\text{adequat number of values}}{\text{total number of values}} = \frac{2^{24}}{2^{28}} = 2^{-4} = 1/16$$