

## Döntéselőkészítés Vizsgafeladat

Unyatszki Dávid  
BGKW2B

Feladatleírás:

Elérkezett a vizsgaidőszak, be kell osztani az időnket, hogy melyik tárgyból mennyit tanulunk és mikor vizsgázunk.

Van 5 tárgy, amiből vizsgázni szeretnénk. Egy tárgyból több vizsgaidőpont is ki van írva. Meg van adva, hogy melyik nap mennyi időt tudunk tanulással tölteni és az is adott, hogy melyik tárgyhoz, mennyi a minimális tanulási idő, hogy sikerüljön a tárgy. Adott a tárgy kredit értéke is.

Továbbá:

- Minden tárgyból vizsgázom és csak egyszer
- Amikor vizsgázom, aznap nem tanulok
- Egymást követő maximum 2 napon keresztül tanulok egy tárgyat, utána másiktól készülök, hogy ne felejtsem el az addig tanultakat

Cél: Lehető legjobb átlag elérése.

Adatok (vizsga.mod fájl):

### # Data

```
set Targyak; // Felvett tárgyak a félévben
```

```
set Napok:= 1..49; //A vizsgaidőszak 7 hétig tart (49 nap)
```

```
param szabadIdo {Napok}; // Egy nap ennyi időnk van tanulni (óra)
```

```
param tanulasIdo {Targyak}; // Azt fejezi ki, hogy mennyit kell tanulni a tárgyból, a jegyek eléréséhez  
(alapból az egyeshez, viszont a kétszerese a ketteshez és így tovább) (óra)
```

```
param vizsgak {Napok, Targyak}; // Vizsgaidőpontok
```

```
param kredit {Targyak}; // A tárgyak kredit értéke
```

### # Variables / Változók

```
var tanul {Targyak, Napok} binary; // A tárgyból aznap tanulok
```

```
var nemTanul {Targyak, Napok} binary; // A tárgyból aznap nem tanulok (még nem vizsgáztam belőle)
```

```
var vizsgazik {Napok, Targyak} binary; // A tárgyból aznap vizsgázom
```

```
var nemTanulLevizsgazott {Targyak, Napok} binary; // A tárgyból aznap nem tanulok, levizsgáztam
```

```
var tanultOrak {Targyak} >=0; // Egy tárgyból mennyit tanultam összesen
```

```
var kapottJegy {Targyak} >=0 integer <=5; // Milyen jegyet kaptam a tárgyból
```

```
var atlag >=0; // Vizsgaidőszak végén az elért átlag
```

### # Constraints / Megkötések

```
# Csak vizsganapon lehet vizsgázni
```

```
s.t. Vizsganap {t in Targyak, n in Napok}:
```

```
    vizsgazik[n,t] <= 1 - 1*(1-vizsgak[n,t]);
```

# Egy tárgyból legfeljebb egyszer vizsgázom

s.t. EgyVizsga {t in Targyak}:

$\text{sum } \{n \text{ in Napok}\} \text{vizsgazik}[n,t]=1;$

# Miután vizsgáztam, a többi napon már nem tanulok az adott tárgyból

s.t. VizsgaUtanNemTanulokATargybol {t in Targyak, n in 2..49}:

$\text{nemTanulLevizsgazott}[t,n] \geq \text{vizsgazik}[n-1,t];$

s.t. NemTanulVizsgaUtan2 {t in Targyak, n in 2..49}:

$\text{nemTanulLevizsgazott}[t,n] \geq \text{nemTanulLevizsgazott}[t,n-1];$

# Egy nap csak egy tárgyból készülök, vagy vizsgázom

s.t. EgyNapEgyTargybolTanulasVagyVizsga {n in Napok, t in Targyak}:

$\text{tanul}[t,n] + \text{vizsgazik}[n,t] + \text{nemTanulLevizsgazott}[t,n] + \text{nemTanul}[t,n] = 1;$

# Minden nap, vagy tanulok, vagy vizsgázom

s.t. MindenNapCsinalokValamit {n in Napok}:

$\text{sum } \{t \text{ in Targyak}\} (\text{tanul}[t,n] + \text{vizsgazik}[n,t]) \leq 1;$

# Egy tárgyat max 2 egymást követő napon tanulok

s.t. Valtozatossag {t in Targyak, n in 3..49}:

$\text{tanul}[t,n] \leq 0 + 1 * (2 - \text{tanul}[t,n-1] - \text{tanul}[t,n-2]);$

# Ha vizsgázok, akkor az nap nem tanulok (lehet nem kell az első miatt)

s.t. HaVizsgazomNemTanulok {t in Targyak, n in Napok}:

$\text{tanul}[t,n] \leq 0 + 1 * (1 - \text{vizsgazik}[n,t]);$

# Tanult órák

s.t. TanultOrak {t in Targyak}:

$\text{tanultOrak}[t] = \text{sum } \{n \text{ in Napok}\} \text{tanul}[t,n] * \text{szabadido}[n];$

# Milyen jegyet kapok a vizsgán, a tanult órák alapján

s.t. KapottJegy {t in Targyak}:

$\text{kapottJegy}[t] \leq \text{tanultOrak}[t] / \text{tanulasido}[t];$

# Átlag (jegy\*kredit/össz kredit)

s.t. AtlagKiszamolas:

$\text{atlag} = (\text{sum } \{t \text{ in Targyak}\} \text{kapottJegy}[t] * \text{kredit}[t]) / (\text{sum } \{t \text{ in Targyak}\} \text{kredit}[t]);$

# Objective/Cél

# Minél jobb átlag elérése

maximize Atlag:

atlag;

solve;

### Kiíratás:

```
printf "\nVizsgaidőszak:\n-----\n";
printf "Átlag: %d\n\n",atlag;
printf "Jegyek:\n";
for {t in Targyak: kapottJegy[t]>=0}
{
    printf "%s: %d (%d tanult óra)\n",t,kapottJegy[t], tanultOrak[t];
}
printf "\n";
for {n in Napok}
{
    printf "%d. napon:\n",n;
    for {t in Targyak: tanul[t,n]>0}
    {
        printf "Tanul: %s\n",t;
    }
    for {t in Targyak: vizsgazik[n,t]>0}
    {
        printf "Vizsgazik: %s\n",t;
    }
}
```

vizsga.dat fájl

set Targyak:= matek prog jog info szoftver;

param szabadIdo:=

1	6.2
2	3.5
3	3
4	6
5	8
6	4
7	0
8	3.4
9	3
10	2.4
11	3
12	3.7
13	3.8
14	0
15	3
16	4
17	6.5
18	8
19	6.4
20	3.4
21	0

```
22    4.3
23    3.6
24    2.8
25    6.3
26    3
27    3
28    0
29    3.4
30    4.2
31    2.6
32    5.3
33    7
34    4
35    0
36    4.3
37    4
38    5
39    4.5
40    3.2
41    4
42    0
43    3
44    3.5
45    4.5
46    3.5
47    3
48    1
49    0
;
```

`param` tanulasido:=

```
matek  8
prog   7
jog     6
info    9
szoftver 10
;
```

`param` vizsgak: matek prog jog info szoftver:=

```
1  0  1  0  0  0
2  1  0  0  0  0
3  0  0  0  1  0
4  0  0  1  0  0
5  0  0  0  0  1
6  0  0  0  0  0
7  0  0  0  0  0
8  0  1  0  0  0
9  1  0  0  0  0
```

10	0	0	0	1	0
11	0	0	1	0	0
12	0	0	0	0	1
13	0	0	0	0	0
14	0	0	0	0	0
15	0	1	0	0	0
16	1	0	0	0	0
17	0	0	0	1	0
18	0	0	1	0	0
19	0	0	0	0	1
20	0	0	0	0	0
21	0	0	0	0	0
22	0	1	0	0	0
23	1	0	0	0	0
24	0	0	0	1	0
25	0	0	1	0	0
26	0	0	0	0	1
27	0	0	0	0	0
28	0	0	0	0	0
29	0	1	0	0	0
30	1	0	0	0	0
31	0	0	0	1	0
32	0	0	1	0	0
33	0	0	0	0	1
34	0	0	0	0	0
35	0	0	0	0	0
36	0	1	0	0	0
37	1	0	0	0	0
38	0	0	0	1	0
39	0	0	1	0	0
40	0	0	0	0	1
41	0	0	0	0	0
42	0	0	0	0	0
43	0	1	0	0	0
44	1	0	0	0	0
45	0	0	0	1	0
46	0	0	1	0	0
47	0	0	0	0	1
48	0	0	0	0	0
49	0	0	0	0	0

;

param kredit:=

matek 4  
prog 3  
jog 2  
info 4  
szoftver 3

