

Zadatci za vježbu

Zadatak 1.

a.

```
File Edit Format Run Options Windows Help
# 01_01_a
class Kvadrat:
    def __init__(self, a = 1):
        self.a = a
        return

    def opseg(self):
        return self.a * 4

    def površina(self):
        return self.a ** 2

a = int(input())
k = Kvadrat(a)
print(k.opseg())
print(k.površina())
```

b.

```
File Edit Format Run Options Windows Help
# 01_01_b
from turtle import *
class Kvadrat:
    def __init__(self, a = 1):
        self.a = a
        return

    def opseg(self):
        return self.a * 4

    def površina(self):
        return self.a ** 2

    def crtaj(self, boja):
        pu()
        bk(self.a // 2)
        rt(90)
        fd(self.a // 2)
        lt(90)
        pd()
        color(boja, boja)
        begin_fill()
        for i in range(4):
            fd(self.a)
            lt(90)
        end_fill()
        mainloop()
        return

a = int(input())
b = input()
k = Kvadrat(a)
k.crtaj(b)
```

Zadatak 2.

a.

```
File Edit Format Run Options Windows Help
# 01_02_a
class Tocka:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
        return

    def udaljenost(self, t):
        return ((self.x - t.x) ** 2 + (self.y - t.y) ** 2) ** 0.5

class Trokut:
    def __init__(self, a = Tocka(0, 0), b = Tocka(0, 0), c = Tocka(0, 0)):
        self.a = a
        self.b = b
        self.c = c
        return

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())

a = Tocka(x1, y1)
b = Tocka(x2, y2)
print(a.udaljenost(b))
```

b.

```
File Edit Format Run Options Windows Help
# 01_02_b
from turtle import *
class Tocka:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
        return

    def udaljenost(self, t):
        return ((self.x - t.x) ** 2 + (self.y - t.y) ** 2) ** 0.5

    def duzina(self, t):
        pencolor('blue')
        pensize(2)
        pu()
        goto(self.x, self.y)
        pd()
        goto(t.x, t.y)
        return

class Trokut:
    def __init__(self, a = Tocka(0, 0), b = Tocka(0, 0), c = Tocka(0, 0)):
        self.a = a
        self.b = b
        self.c = c
        return

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())

a = Tocka(x1, y1)
b = Tocka(x2, y2)
a.duzina(b)
mainloop()
```

C.

```
File Edit Format Run Options Windows Help
# 01_02_c
from turtle import *
class Tocka:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
        return

    def udaljenost(self, t):
        return ((self.x - t.x) ** 2 + (self.y - t.y) ** 2) ** 0.5

    def duzina(self, t):
        pencolor('blue')
        pensize(2)
        pu()
        goto(self.x, self.y)
        pd()
        goto(t.x, t.y)
        return

class Trokut:
    def __init__(self, a = Tocka(0, 0), b = Tocka(0, 0), c = Tocka(0, 0)):
        self.a = a
        self.b = b
        self.c = c
        return

    def opseg(self):
        return a.udaljenost(b) + a.udaljenost(c) + b.udaljenost(c)

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
x3 = int(input())
y3 = int(input())

a = Tocka(x1, y1)
b = Tocka(x2, y2)
c = Tocka(x3, y3)
t = Trokut(a, b, c)
print('{:.2f}'.format(t.opseg()))
```

d.

```

File Edit Format Run Options Windows Help
# 01_02_d
from turtle import *
class Tocka:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
        return

    def udaljenost(self, t):
        return ((self.x - t.x) ** 2 + (self.y - t.y) ** 2) ** 0.5

    def duzina(self, t):
        pencolor('blue')
        pensize(2)
        pu()
        goto(self.x, self.y)
        pd()
        goto(t.x, t.y)
        return

class Trokut:
    def __init__(self, a = Tocka(0, 0), b = Tocka(0, 0), c = Tocka(0, 0)):
        self.a = a
        self.b = b
        self.c = c
        return

    def opseg(self):
        return a.udaljenost(b) + a.udaljenost(c) + b.udaljenost(c)

    def površina(self):
        s = self.opseg() / 2
        ab = a.udaljenost(b)
        ac = a.udaljenost(c)
        bc = b.udaljenost(c)
        return (s * (s - ab) * (s - ac) * (s - bc)) ** 0.5

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
x3 = int(input())
y3 = int(input())

a = Tocka(x1, y1)
b = Tocka(x2, y2)
c = Tocka(x3, y3)
t = Trokut(a, b, c)
print('{:.2f}'.format(t.površina()))

```

Zadatak 3.

a.

```
File Edit Format Run Options Windows Help
# 01_03_a
class Pravac:
    def __init__(self, a = 0, b = 0, c = 0):
        self.A = a
        self.B = b
        self.C = c
        return

    def paralelni(self, p):
        if self.B == 0:
            if p.B == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == p.A / p.B

A = int(input())
B = int(input())
C = int(input())
p1 = Pravac(A, B, C)
A = int(input())
B = int(input())
C = int(input())
p2 = Pravac(A, B, C)
if p1.paralelni(p2):
    print('Pravci su paralelni')
else:
    print('Pravci nisu paralelni')
```

b.

```
File Edit Format Run Options Windows Help
# 01_03_b
class Pravac:
    def __init__(self, a = 0, b = 0, c = 0):
        self.A = a
        self.B = b
        self.C = c
        return

    def paralelni(self, p):
        if self.B == 0:
            if p.B == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == p.A / p.B

    def okomiti(self, p):
        if self.B == 0:
            if p.A == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == -p.B / p.A

A = int(input())
B = int(input())
C = int(input())
p1 = Pravac(A, B, C)
A = int(input())
B = int(input())
C = int(input())
p2 = Pravac(A, B, C)
if p1.okomiti(p2):
    print('Pravci su okomiti')
else:
    print('Pravci nisu okomiti')
```


C.

```
File Edit Format Run Options Windows Help
# 01_03_c
class Pravac:
    def __init__(self, a = 0, b = 0, c = 0):
        self.A = a
        self.B = b
        self.C = c
        return

    def paralelni(self, p):
        if self.B == 0:
            if p.B == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == p.A / p.B

    def okomiti(self, p):
        if self.B == 0:
            if p.A == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == -p.B / p.A

    def sjeciste(self, p):
        if not self.paralelni(p):
            x = (p.C * self.B - self.C * p.B) / (self.A * p.B - p.A * self.B)
            y = (p.A * self.B - self.A * p.C) / (self.A * p.B - p.A * self.B)
            return x, y
        else:
            return None

A = int(input())
B = int(input())
C = int(input())
p1 = Pravac(A, B, C)
A = int(input())
B = int(input())
C = int(input())
p2 = Pravac(A, B, C)
t = p1.sjeciste(p2)
if t != None:
    print(t)
else:
    print('Pravci se ne sijeku')
```

d.

```

File Edit Format Run Options Windows Help
# 01_03_d
from turtle import *
class Pravac:
    def __init__(self, a = 0, b = 0, c = 0):
        self.A = a
        self.B = b
        self.C = c
        return

    def paralelni(self, p):
        if self.B == 0:
            if p.B == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == p.A / p.B

    def okomiti(self, p):
        if self.B == 0:
            if p.A == 0:
                return True
            else:
                return False
        else:
            return self.A / self.B == -p.B / p.A

    def sjeciste(self, p):
        if not self.paralelni(p):
            x = (p.C * self.B - self.C * p.B) / (self.A * p.B - p.A * self.B)
            y = (p.A * self.B - self.A * p.C) / (self.A * p.B - p.A * self.B)
            return x, y
        else:
            return None

    def okvir(self, x_dim, y_dim):
        ht(); pu(); speed(0)
        x_p = -x_dim // 2; y_p = -y_dim // 2
        goto(x_p, y_p)
        pd(); pencolor('black')
        fd(x_dim); lt(90); fd(y_dim); lt(90); fd(x_dim); lt(90); fd(y_dim); lt(90)
        return

    def podjela_osi(self, os_dim, oznake):
        pd = len(oznake) - 1
        d = round(os_dim / pd)
        razlika = os_dim - d * pd
        os = [-os_dim // 2]
        for i in range(1, pd + 1):
            os.append(os[i-1] + d)
            if razlika != 0 and (i % (pd // abs(razlika)) == 0):
                if razlika > 0:

```

```

        os[i] += 1
    else:
        os[i] -= 1
    return os

def crtanje_x_osi(self, x_dim, y_dim, oz_x, opis_x, mreža):
    x_os = self.podjela_osi(x_dim, oz_x)
    seth(90)
    y_p = -y_dim // 2
    for i in range(1, len(x_os) - 1):
        pu(); goto(x_os[i], y_p)
        pd(); fd(5)
        if mreža == 1:
            pencolor(0.8, 0.8, 0.8)
            fd(y_dim - 5)
            pencolor('black')
    for i in range(len(x_os)):
        pu(); goto(x_os[i], y_p - 20)
        write(oz_x[i], align = 'center')
    pu(); goto(x_dim // 2 - len(opis_x), y_p - 40)
    write(opis_x, align = 'right', font = ('Arial', 10, 'normal'))
    return

def crtanje_y_osi(self, x_dim, y_dim, oz_y, opis_y, mreža):
    y_os = self.podjela_osi(y_dim, oz_y)
    seth(0)
    x_p = -x_dim // 2; y_p = -y_dim // 2
    for i in range(1, len(y_os) - 1):
        pu(); goto(x_p, y_os[i])
        pd(); fd(5)
        if mreža == 1:
            pencolor(0.8, 0.8, 0.8)
            fd(x_dim - 5)
            pencolor('black')
    for i in range(len(y_os)):
        pu(); goto(x_p - 8 - len(str(oz_y[i])), y_os[i] - 7)
        write(oz_y[i], align = 'right')
    pu(); goto(x_p - 8 - len(opis_y), y_dim // 2 + 10)
    write(opis_y, align = 'center', font = ('Arial', 10, 'normal'))
    return

def koord_sustav(self, oz_x, oz_y, op_x, op_y, x_dim, y_dim, mreža):
    self.okvir(x_dim, y_dim)
    self.crtanje_x_osi(x_dim, y_dim, oz_x, op_x, mreža)
    self.crtanje_y_osi(x_dim, y_dim, oz_y, op_y, mreža)
    pu(); goto(0, y_dim // 2 + 20)
    return

def crtaj(self, d):
    s = 400
    v = 400
    oz_x = []
    for t in range(-s // d // 2, s // d // 2 + 1):

```

```
        oz_x.append(t)
    oz_y = []
    for t in range(-v // d // 2, v // d // 2 + 1):
        oz_y.append(t)
    self.koord_sustav(oz_x, oz_y, 'x', 'y', s, v, 1)
    x0 = -s // d // 2
    pencolor('red')
    if self.B != 0:
        y0 = -self.A * x0 / self.B - self.C / self.B
        pu()
        goto(x0 * d, y0 * d)
        pd()
        while x0 <= s // d // 2:
            y0 = -self.A * x0 / self.B - self.C / self.B
            if y0 < -v // d // 2 or y0 > v // d // 2:
                pu()
            else:
                pd()
            goto(x0 * d, y0 * d)
            x0 += 1
    else:
        if self.A != 0:
            pu()
            goto(-s // 2, -self.C / self.A * d)
            pd()
            goto(s // 2, -self.C / self.A * d)
    return

A = int(input())
B = int(input())
C = int(input())
d = int(input())
p1 = Pravac(A, B, C)
p1.crtaj(d)
```

Zadatak 4.

a.

```
File Edit Format Run Options Windows Help
# 01_04_a
class Vrijeme:
    def __init__(self, h, m):
        self.H = h
        self.M = m
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        return '{}:{}'.format(h, m)

    def __lt__(self, v):
        return self.H * 60 + self.M < v.H * 60 + v.M

    def __eq__(self, v):
        return self.H == v.H and self.M == v.M

    def __gt__(self, v):
        return self.H * 60 + self.M > v.H * 60 + v.M

h = int(input())
m = int(input())
v = Vrijeme(h, m)
print(v)
```

b.

```
File Edit Format Run Options Windows Help
# 01_04 b
class Vrijeme:
    def __init__(self, h, m):
        self.H = h
        self.M = m
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        return '{}:{}'.format(h, m)

    def __lt__(self, v):
        return self.H * 60 + self.M < v.H * 60 + v.M

    def __eq__(self, v):
        return self.H == v.H and self.M == v.M

    def __gt__(self, v):
        return self.H * 60 + self.M > v.H * 60 + v.M

h = int(input())
m = int(input())
v1 = Vrijeme(h, m)
h = int(input())
m = int(input())
v2 = Vrijeme(h, m)
if v1 == v2:
    print('Jednako')
elif v1 > v2:
    print('Vece')
else:
    print('Manje')
```

C.

```
File Edit Format Run Options Windows Help
# 01_04_c
class Vrijeme:
    def __init__(self, h, m):
        self.H = h
        self.M = m
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        return '{}:{}'.format(h, m)

    def __lt__(self, v):
        return self.H * 60 + self.M < v.H * 60 + v.M

    def __eq__(self, v):
        return self.H == v.H and self.M == v.M

    def __gt__(self, v):
        return self.H * 60 + self.M > v.H * 60 + v.M

    def __add__(self, m):
        v = self.H * 60 + self.M + m
        return Vrijeme(v // 60, v % 60)

h = int(input())
m = int(input())
v1 = Vrijeme(h, m)
m = int(input())
v2 = v1 + m
print(v2)
```

d.

```
File Edit Format Run Options Windows Help
# 01_04_d
class Vrijeme:
    def __init__(self, h, m):
        self.H = h
        self.M = m
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        return '{}:{}'.format(h, m)

    def __lt__(self, v):
        return self.H * 60 + self.M < v.H * 60 + v.M

    def __eq__(self, v):
        return self.H == v.H and self.M == v.M

    def __gt__(self, v):
        return self.H * 60 + self.M > v.H * 60 + v.M

    def __add__(self, m):
        v = self.H * 60 + self.M + m
        return Vrijeme(v // 60, v % 60)

    def __sub__(self, v):
        k = abs((self.H - v.H) * 60 + self.M - v.M)
        return Vrijeme(k // 60, k % 60)

h = int(input())
m = int(input())
v1 = Vrijeme(h, m)
h = int(input())
m = int(input())
v2 = Vrijeme(h, m)
print(v2 - v1)
```


Zadatak 5.

a.

```
File Edit Format Run Options Windows Help
# 01_05_a
class Cetverokut:
    def __init__(self, a, b, c, d):
        self.a = a
        self.b = b
        self.c = c
        self.d = d
        return

    def opseg(self):
        return self.a + self.b + self.c + self.d

a = int(input())
b = int(input())
c = int(input())
d = int(input())
cet = Cetverokut(a, b, c, d)
print(cet.opseg())
```

b.

```
File Edit Format Run Options Windows Help
# 01_05_b
class Cetverokut:
    def __init__(self, a, b, c, d):
        self.a = a
        self.b = b
        self.c = c
        self.d = d
        return

    def opseg(self):
        return self.a + self.b + self.c + self.d

class Pravokutnik(Cetverokut):
    def __init__(self, a, b):
        super().__init__(a, b, a, b)
        return

    def površina(self):
        return self.a * self.b

a = int(input())
b = int(input())
p = Pravokutnik(a, b)
print(p.površina())
```

c.

```
File Edit Format Run Options Windows Help
# 01_05_c
class Cetverokut:
    def __init__(self, a, b, c, d):
        self.a = a
        self.b = b
        self.c = c
        self.d = d
        return

    def opseg(self):
        return self.a + self.b + self.c + self.d

class Pravokutnik(Cetverokut):
    def __init__(self, a, b):
        super().__init__(a, b, a, b)
        return

    def površina(self):
        return self.a * self.b

    def kvadrat(self):
        return self.a == self.b

a = int(input())
b = int(input())
p = Pravokutnik(a, b)
if p.kvadrat():
    print('Kvadrat')
else:
    print('Nije kvadrat')
```

Zadatak 6.

a.

```
File Edit Format Run Options Windows Help
# 01_06_a
class Vrijeme:
    def __init__(self, h, m, s):
        self.H = h
        self.M = m
        self.S = s
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        s = str(self.S)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        if len(s) < 2:
            s = '0' + s
        return '{}: {}: {}'.format(h, m, s)

    def __lt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S < v.H * 3600 + v.M * 60 + v.S

    def __gt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S > v.H * 3600 + v.M * 60 + v.S

class Emisija:
    def __init__(self, naziv, pocetak, kraj):
        self.N = naziv
        self.P = pocetak
        self.K = kraj
        return

h = int(input())
m = int(input())
s = int(input())
v1 = Vrijeme(h, m, s)
h = int(input())
m = int(input())
s = int(input())
v2 = Vrijeme(h, m, s)
if v1 < v2:
    print(v1)
else:
    print(v2)
```

b.

```

File Edit Format Run Options Windows Help
# 01_06_b
class Vrijeme:
    def __init__(self, h, m, s):
        self.H = h
        self.M = m
        self.S = s
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        s = str(self.S)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        if len(s) < 2:
            s = '0' + s
        return '{}: {}: {}'.format(h, m, s)

    def __lt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S < v.H * 3600 + v.M * 60 + v.S

    def __gt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S > v.H * 3600 + v.M * 60 + v.S

    def __sub__(self, v):
        t = (self.H - v.H) * 3600 + (self.M - v.M) * 60 + self.S - v.S
        h = t // 3600
        t = t % 2600
        m = t // 60
        s = t % 60
        return Vrijeme(h, m, s)

class Emisija:
    def __init__(self, naziv, pocetak, kraj):
        self.N = naziv
        self.P = pocetak
        self.K = kraj
        return

    def __repr__(self):
        return '{:20} {:10} {:10}'.format(self.N, self.P, self.K)

    def __lt__(self, e):
        return self.K - self.P < e.K - e.P

    def __gt__(self, e):
        return self.K - self.P > e.K - e.P

naziv = input()
h = int(input())
m = int(input())

```

```
s = int(input())
v1 = Vrijeme(h, m, s)
h = int(input())
m = int(input())
s = int(input())
v2 = Vrijeme(h, m, s)
e1 = Emisija(naziv, v1, v2)
naziv = input()
h = int(input())
m = int(input())
s = int(input())
v1 = Vrijeme(h, m, s)
h = int(input())
m = int(input())
s = int(input())
v2 = Vrijeme(h, m, s)
e2 = Emisija(naziv, v1, v2)
if e1 > e2:
    print(e1)
else:
    print(e2)
```

C.

```

File Edit Format Run Options Windows Help
# 01_06_c
class Vrijeme:
    def __init__(self, h, m, s):
        self.H = h
        self.M = m
        self.S = s
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        s = str(self.S)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        if len(s) < 2:
            s = '0' + s
        return '{}: {}: {}'.format(h, m, s)

    def __lt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S < v.H * 3600 + v.M * 60 + v.S

    def __gt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S > v.H * 3600 + v.M * 60 + v.S

    def __sub__(self, v):
        t = (self.H - v.H) * 3600 + (self.M - v.M) * 60 + self.S - v.S
        h = t // 3600
        t = t % 2600
        m = t // 60
        s = t % 60
        return Vrijeme(h, m, s)

class Emisija:
    def __init__(self, naziv, pocetak, kraj):
        self.N = naziv
        self.P = pocetak
        self.K = kraj
        return

    def __repr__(self):
        return '{:20} {:10} {:10}'.format(self.N, self.P, self.K)

    def __lt__(self, e):
        return self.K - self.P < e.K - e.P

    def __gt__(self, e):
        return self.K - self.P > e.K - e.P

class Program:
    def __init__(self, naziv, emisije = []):
        self.N = naziv

```

```
        self.E = emisije
        return

    def __repr__(self):
        s = self.N + '\n'
        s += '=' * 40 + '\n'
        for t in self.E:
            s += str(t) + '\n'
        return s

    def dodajEmisiju(self, e):
        self.E.append(e)
        return

    def sortirajN(self):
        self.E = sorted(self.E, key = lambda e : e.N)
        return

naziv = input()
n = int(input())
p = Program(naziv)
for i in range(n):
    naziv = input()
    h = int(input())
    m = int(input())
    s = int(input())
    v1 = Vrijeme(h, m, s)
    h = int(input())
    m = int(input())
    s = int(input())
    v2 = Vrijeme(h, m, s)
    e = Emisija(naziv, v1, v2)
    p.dodajEmisiju(e)
p.sortirajN()
print(p)
```

d.

```

File Edit Format Run Options Windows Help
# 01_06_d
class Vrijeme:
    def __init__(self, h, m, s):
        self.H = h
        self.M = m
        self.S = s
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        s = str(self.S)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        if len(s) < 2:
            s = '0' + s
        return '{}: {}: {}'.format(h, m, s)

    def __lt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S < v.H * 3600 + v.M * 60 + v.S

    def __gt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S > v.H * 3600 + v.M * 60 + v.S

    def __sub__(self, v):
        t = (self.H - v.H) * 3600 + (self.M - v.M) * 60 + self.S - v.S
        h = t // 3600
        t = t % 2600
        m = t // 60
        s = t % 60
        return Vrijeme(h, m, s)

class Emisija:
    def __init__(self, naziv, pocetak, kraj):
        self.N = naziv
        self.P = pocetak
        self.K = kraj
        return

    def __repr__(self):
        return '{:20} {:10} {:10}'.format(self.N, self.P, self.K)

    def __lt__(self, e):
        return self.K - self.P < e.K - e.P

    def __gt__(self, e):
        return self.K - self.P > e.K - e.P

class Program:
    def __init__(self, naziv, emisije = []):
        self.N = naziv

```



```
        self.E = emisije
        return

    def __repr__(self):
        s = self.N + '\n'
        s += '=' * 40 + '\n'
        for t in self.E:
            s += str(t) + '\n'
        return s

    def dodajEmisiju(self, e):
        self.E.append(e)
        return

    def sortirajN(self):
        self.E = sorted(self.E, key = lambda e : e.N)
        return

    def sortirajV(self):
        self.E = sorted(self.E, key = lambda e : e.P)
        return

naziv = input()
n = int(input())
p = Program(naziv)
for i in range(n):
    naziv = input()
    h = int(input())
    m = int(input())
    s = int(input())
    v1 = Vrijeme(h, m, s)
    h = int(input())
    m = int(input())
    s = int(input())
    v2 = Vrijeme(h, m, s)
    e = Emisija(naziv, v1, v2)
    p.dodajEmisiju(e)
p.sortirajV()
print(p)
```

e.

```

File Edit Format Run Options Windows Help
# 01_06_e
class Vrijeme:
    def __init__(self, h, m, s):
        self.H = h
        self.M = m
        self.S = s
        return

    def __repr__(self):
        h = str(self.H)
        m = str(self.M)
        s = str(self.S)
        if len(h) < 2:
            h = '0' + h
        if len(m) < 2:
            m = '0' + m
        if len(s) < 2:
            s = '0' + s
        return '{}: {}: {}'.format(h, m, s)

    def __lt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S < v.H * 3600 + v.M * 60 + v.S

    def __gt__(self, v):
        return self.H * 3600 + self.M * 60 + self.S > v.H * 3600 + v.M * 60 + v.S

    def __sub__(self, v):
        t = (self.H - v.H) * 3600 + (self.M - v.M) * 60 + self.S - v.S
        h = t // 3600
        t = t % 2600
        m = t // 60
        s = t % 60
        return Vrijeme(h, m, s)

class Emisija:
    def __init__(self, naziv, pocetak, kraj):
        self.N = naziv
        self.P = pocetak
        self.K = kraj
        return

    def __repr__(self):
        return '{:20} {:10} {:10}'.format(self.N, self.P, self.K)

    def __lt__(self, e):
        return self.K - self.P < e.K - e.P

    def __gt__(self, e):
        return self.K - self.P > e.K - e.P

class Program:
    def __init__(self, naziv, emisije = []):
        self.N = naziv

```

```
        self.E = emisije
        return

    def __repr__(self):
        s = self.N + '\n'
        s += '=' * 40 + '\n'
        for t in self.E:
            s += str(t) + '\n'
        return s

    def dodajEmisiju(self, e):
        self.E.append(e)
        return

    def sortirajN(self):
        self.E = sorted(self.E, key = lambda e : e.N)
        return

    def sortirajV(self):
        self.E = sorted(self.E, key = lambda e : e.P)
        return

    def sortirajT(self):
        self.E = sorted(self.E, key = lambda e : e.K - e.P)
        return

naziv = input()
n = int(input())
p = Program(naziv)
for i in range(n):
    naziv = input()
    h = int(input())
    m = int(input())
    s = int(input())
    v1 = Vrijeme(h, m, s)
    h = int(input())
    m = int(input())
    s = int(input())
    v2 = Vrijeme(h, m, s)
    e = Emisija(naziv, v1, v2)
    p.dodajEmisiju(e)
p.sortirajT()
print(p)
```