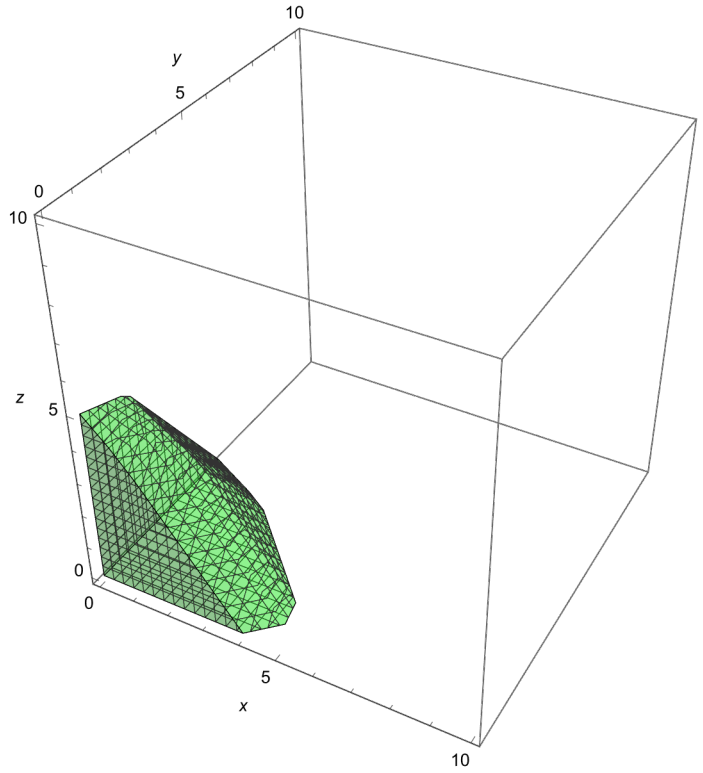
(*
Задача 2. Решете задачата на линейното оптимиране
  F (x,y,z)=x+y+3z→max
    Ω е областта съставена от
  |4x−4y+3z−16≤0
  |4x+4y+3z−24≤0
  |2x+4y+3z−20≤0
  |x+y+3z−15≤0
  |−x+y−z+5≥0
  |x≥0
  |y≥0
  |z≥0

*)

(* 1. Скицирайте областта Ω и декартовата координатна система. *)

In[•]:=
```
p = RegionPlot3D[
  4 * x - 4 * y + 3 * z - 16 ≤ 0 &&
   4 * x + 4 * y + 3 * z - 24 ≤ 0 &&
   2 * x + 4 * y + 3 * z - 20 ≤ 0 &&
   x + y + 3 * z - 15 ≤ 0 &&
   -x + y - z + 5 ≥ 0 &&
   x ≥ 0 &&
   y ≥ 0 &&
   z ≥ 0,
  {x, 0, 10},
  {y, 0, 10},
  {z, 0, 10},
  Axes → True,
  AxesLabel → {x, y, z},
  PlotStyle → {Directive[Green, Opacity[0.25]]}
 ]
```

Out[•]=




```
(* 2. Направете анимация за откриване на оптималните решения. *)
```

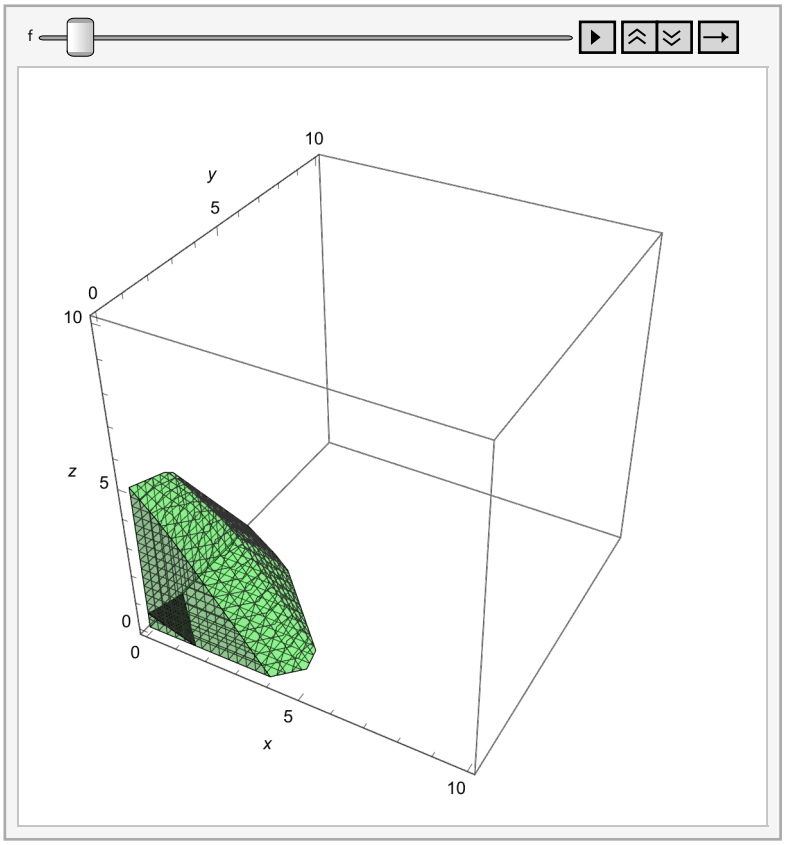
In[•]:=
```
F[x_, y_, z_] = x + y + 3 z
```

Out[•]=

$$x + y + 3\,z$$

In[•]:=
```
Animate[
 Show[
  RegionPlot3D[
   4 * x - 4 * y + 3 * z - 16 ≤ 0 &&
    4 * x + 4 * y + 3 * z - 24 ≤ 0 &&
    2 * x + 4 * y + 3 * z - 20 ≤ 0 &&
    x + y + 3 * z - 15 ≤ 0 &&
    -x + y - z + 5 ≥ 0 &&
    x ≥ 0 &&
    y ≥ 0 &&
    z ≥ 0,
   {x, 0, 10},
   {y, 0, 10},
   {z, 0, 10},
   Axes → True,
   AxesLabel → {x, y, z},
   PlotStyle → {Directive[Green, Opacity[0.25]]}
  ],
  ContourPlot3D[
   x + y + 3 * z == f,
   {x, 0, 10},
   {y, 0, 10},
   {z, 0, 10},
   ContourStyle → Directive[Black, Thin],
   Axes → True,
   AxesLabel → {x, y, z}
  ]
 ],
 {f, 0, 50} ,
 AnimationRunning → False
]
```

Out[•]=




```
(* 3. Намерете едно решение с функцията LinearProgramming[...]. *)
```

```
(*
 |4x-4y+3z≤16
 |4x+4y+3z≤24
 |2x+4y+3z≤20
 |x+y+3z≤15
 |x-y+z≤5
 |x≥0
 |y≥0
 |z≥0
*)
```

In[•]:=
```
F[x_, y_, z_] = x + y + 3 z
```

Out[•]=
```
x + y + 3 z
```

In[•]:=
```
c = {1, 1, 3}
```

Out[•]=
```
{1, 1, 3}
```

In[•]:=
$$m = \begin{pmatrix} 4 & -4 & 3 \\ 4 & 4 & 3 \\ 2 & 4 & 3 \\ 1 & 1 & 3 \\ 1 & -1 & 1 \end{pmatrix}$$

Out[•]=
```
{{4, -4, 3}, {4, 4, 3}, {2, 4, 3}, {1, 1, 3}, {1, -1, 1}}
```

In[•]:=
```
b = {16, 24, 20, 15, 5}
```

Out[•]=
```
{16, 24, 20, 15, 5}
```

In[•]:=
```
v = LinearProgramming[c, m, b]
```

Out[•]=
```
{2, 1, 4}
```

In[•]:=
```
F[2, 1, 4]
```

Out[•]=
```
15
```

In[•]:=
```
{2, 1, 4}
```

Out[•]=
```
{2, 1, 4}
```

In[•]:=
```
Maximize[{F[x, y, z], 4 * x - 4 * y + 3 * z - 16 ≤ 0 &&
   4 * x + 4 * y + 3 * z - 24 ≤ 0 &&
   2 * x + 4 * y + 3 * z - 20 ≤ 0 &&
   x + y + 3 * z - 15 ≤ 0 &&
   -x + y - z + 5 ≥ 0 &&
   x ≥ 0 &&
   y ≥ 0 &&
   z ≥ 0}, {x, y, z}]
```

Out[•]=
```
{15, {x → 0, y → 0, z → 5}}
```

In[•]:=
```
(* При известен максимум търсим всички оптимални планове с функцията Reduce *)
Reduce[
 x + y + 3 z == 15 &&
  4 * x - 4 * y + 3 * z - 16 ≤ 0 &&
  4 * x + 4 * y + 3 * z - 24 ≤ 0 &&
  2 * x + 4 * y + 3 * z - 20 ≤ 0 &&
  x + y + 3 * z - 15 ≤ 0 &&
  -x + y - z + 5 ≥ 0 &&
  x ≥ 0 &&
  y ≥ 0 &&
  z ≥ 0,
 {x, y, z}
]
```

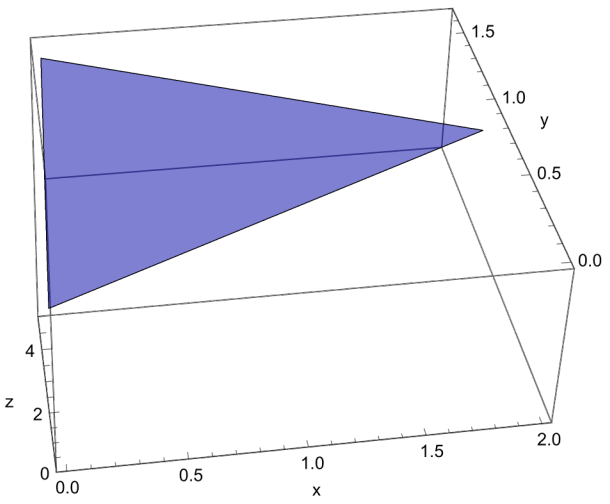
Out[•]=

$$\left(\left(0 \le x < 2 \;\&\&\; \frac{x}{2} \le y \le \frac{5-x}{3}\right) \;||\; (x == 2 \;\&\&\; y == 1)\right) \;\&\&\; z == \frac{1}{3}\,(15 - x - y)$$

```
(* Получаваме всички решения *)
```

In[•]:=
```
region = Plot3D[(1 / 3) (15 - x - y), {x, 0, 2}, {y, x / 2, (5 - x) / 3},
  PlotRange → {Automatic, Automatic, {0, 5}}, Mesh → None, Axes → True,
  AxesLabel → {"x", "y", "z"}, PlotStyle → Directive[Blue, Opacity[0.5]]]
```

Out[•]=



In[•]:= **Show[p, region]**

Out[•]=