Domaći zadatak broj 2

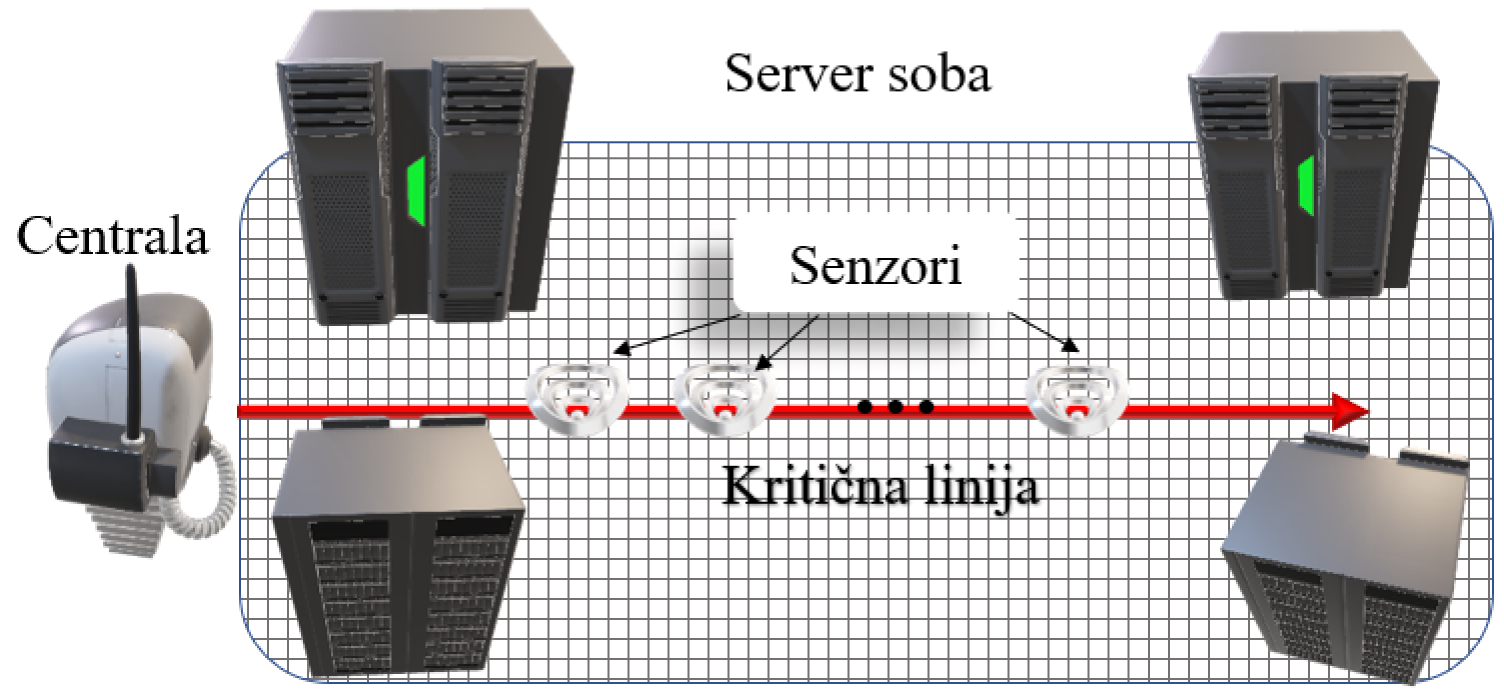
Ime i prezime studenta: Nastasija Stankovic Broj indeksa: 17955

Uputstvo: 1. Pre početka izrade promenite ime datoteke u Domaci2_ime_prezime. (ubacite svoje ime i prezime) 2. Popunite ćeliju ispod naslova odgovarajućim podacima. 3. Prvo rešite prvi zadatak kako bi naredbe modula bile dostupne. 4. Za rešavanje zadataka, ukoliko je potrebno, otvorite ispod teksta zadataka dodatne ćelije za upisivanje tekstualnog odgovora (Markdown) ili programskog koda (Code). 5. Nakon završetka izrade rešenja Notebook dokument sačuvati u pdf formatu i proslediti ga nastavniku. To možete da uradite ili kroz Teams ili na mail adresu jovana.dzunic@elfak.ni.ac.rs

Zadatak 1. Učitati module NumPy, Random, Matplotlib, Pyplot i dati im skraćena imena. Te skraćenice koristiti u pozivu naredbi ovih modula u ostalim zadacima.

(1poen)

In [4]:
```
import numpy as np
import numpy.random as rnd
import matplotlib as mpl
import matplotlib.pyplot as plt
```



Zadatak 2. Za očitavanje temperature u server sobi postavljeno je 14 senzora ekvidistantno po kritičnoj liniji na udaljenosti od 2m do 8m od centrale. Rezultati očitavanja nalaze se u vektoru temperature.

Kreirati vektor temperature dimenzije 14 celobronih slučajnih vrednosti u opsegu $[15, 35]$ (obe granice dolaze u obzir).

(3 poena)

In [5]:
```
temperatura=rnd.randint(15,36,size=14)
print(temperatura)
```

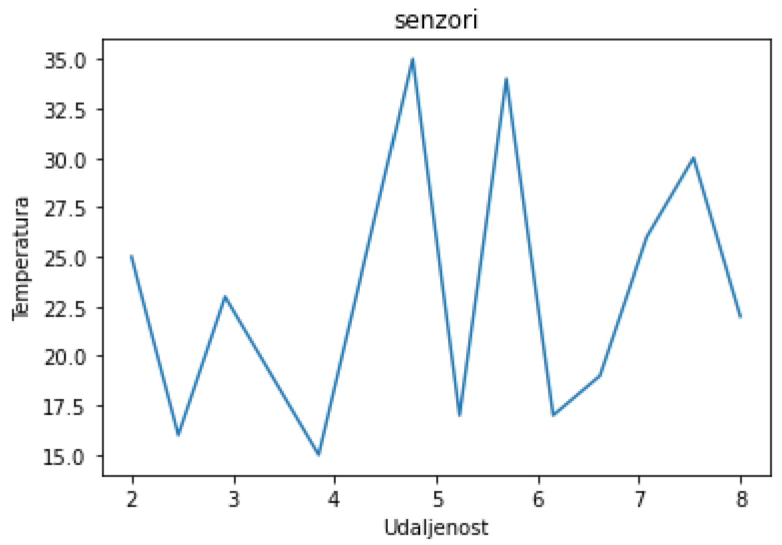
```
[25 16 23 19 15 25 35 17 34 17 19 26 30 22]
```

Napraviti linijski grafički prikaz vrednosti (x_k, y_k) tako da $x-$koordinata odgovara udaljenosti senzora od centrale (14 ekvidistantnih vrednosti od 2m do 8m), $y-$kooridnata odgovara izmerenoj temperaturi. Dati odgovarajuća imena koordinatnim osama, npr. udaljenost i temperatura. Nasloviti grafik: senzori.

(6 poena)

In [10]:
```
udaljenost = np.linspace(2,8,num=14)
plt.plot(udaljenost,temperatura)
plt.xlabel("Udaljenost")
plt.ylabel("Temperatura")
plt.title("senzori")
```

Out[10]:
```
Text(0.5, 1.0, 'senzori')
```



Zadatak 3. Za vektor $v \in \mathbb{R}^n$ kažemo da je nenegativan ukoliko su sve njegove komponente nenegativni brojevi. To ćemo označavati kraće $v \geq 0$. Pokazati da za $v, u \in \mathbb{R}^n$, $v, u \geq 0$ važi da je skalarni proizvod $v \cdot u \geq 0$.

(5 poena) Odgovor na ovo pitanje je tekstualan.

Neka su dati vektori v i u:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

i pretpostavimo da za njih važi $v_i, u_i \geq 0$.

Skalarni proizvod ovih vektora jednak je: $v_1u_1 + v_2u_2 + \cdots + v_nu_n$.

Kako za komponente vektora v i u važi $v, u \geq 0$, može se zaključiti da je svaki proizvod $v_iu_i \geq 0$.

Zbir svih ovih nenegativnih proizvoda komponenti vektora v i u je takođe nenegativna, pa zaključujemo da za ove vektore važi da je njihov skalarni proizvod $v \cdot u \geq 0$.

Opisati situaciju kada za nenegativne vektore $v \geq 0$ i $u \geq 0$ važi da je $v \cdot u = 0$. Vaš odgovor treba da bude u kontekstu šablona pojavljivanja nula među komponentama ovih vektora.

(5 poena) Odgovor na ovo pitanje je tekstualan.

Posto je skalarni proizvod dva vektora :

$$v \cdot u = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = v_1u_1 + v_2u_2 + \cdots + v_nu_n.$$

Dakle, da bi skalarni proizvod ovih vektora bio jednak nuli, na svakom indeksu se mora pojaviti vrednost 0 kao komponenta barem jednog od ovih vektora. Primer:

$$v = \begin{bmatrix} 0 \\ 3 \\ 0 \\ 9 \end{bmatrix}$$

$$u = \begin{bmatrix} 4 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

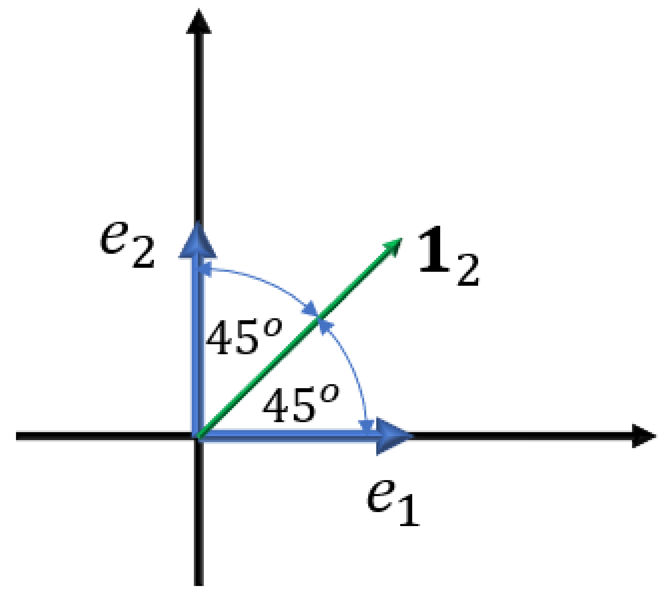
Zadatak 4. Uticaj povećanja dimezije vektorskog prostora na ugao između vektora $\mathbf{1}_n \in \mathbb{R}^n$ i vektora kanonske baze $\{e_1, e_2, \ldots. e_n\}$ prostora $\mathbb{R}^n$:

$$\mathbf{1}_n = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T$$

Za odgovor na ovo pitanje koristiti Python kod za izračunavanja, a komentar je tekstualan.

a) U prostoru $\mathbb{R}^2$, posmatrano u odnosu na Dekartov koordinatni sistem, linija koja polovi prvi kvadrant tj. vektor $\mathbf{1}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ gradi uglove od 45^o sa svakom od koordinatnih osa. Koordinatne ose određene su kanonskom bazom

$$e_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \qquad e_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T.$$



Pokazati ovo tvrđenje izračunavanjem kosinusa odgovarajućih uglova.

(2 poena)

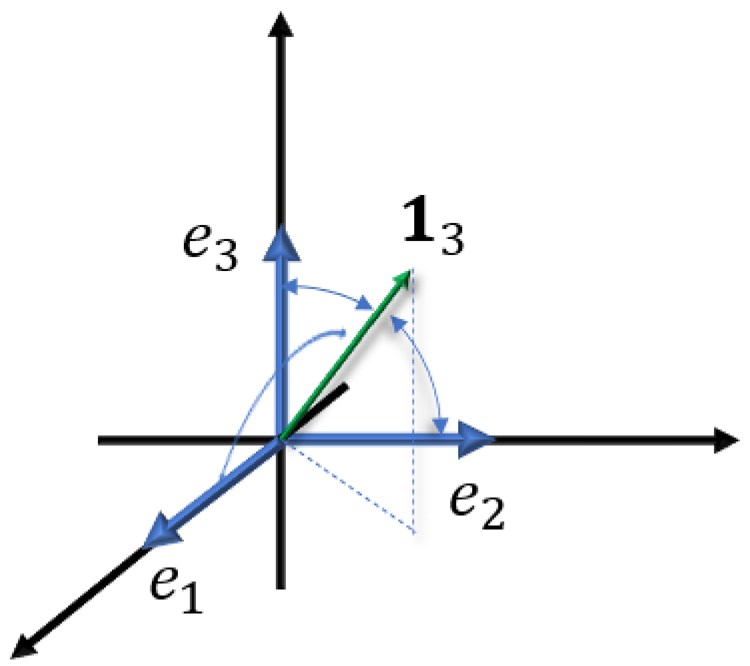
In [13]:
```
m1=np.array([1,1])
m2=np.array([1,0])
m3=np.array([0,1])
```

```
cosinus1 = np.dot(m1,m2)/(np.linalg.norm(m1)*np.linalg.norm(m2))
cosinus2 = np.dot(m1,m3)/(np.linalg.norm(m1)*np.linalg.norm(m3))
print(cosinus1)
print(cosinus2)
```

```
0.7071067811865475
0.7071067811865475
```

b) U prostoru $\mathbb{R}^3$ koje uglove zaklapa vektor $\mathbf{1}_3 = [1 \quad 1 \quad 1]^T$ (dijagonalni vektor prvog oktanta) sa svakim od vektora kanonske baze

$$e_1 = [1 \quad 0 \quad 0]^T, \qquad e_2 = [0 \quad 1 \quad 0]^T, \qquad e_3 = [0 \quad 0 \quad 1]^T?$$



(2 poena)

In [15]:
```
m1=np.array([1,0,0])
m2=np.array([0,1,0])
m3=np.array([0,0,1])
m4=np.array([1,1,1])
cosinus1=np.dot(m4,m1)/(np.linalg.norm(m4))*(np.linalg.norm(m1))
cosinus2=np.dot(m4,m2)/(np.linalg.norm(m4))*(np.linalg.norm(m2))
cosinus3=np.dot(m4,m3)/(np.linalg.norm(m4))*(np.linalg.norm(m3))
p1=np.arccos(cosinus1)
p2=np.arccos(cosinus2)
p3=np.arccos(cosinus3)
print (np.degrees(p1))
print (np.degrees(p2))
print (np.degrees(p3))
```

```
54.735610317245346
54.735610317245346
54.735610317245346
```

c) Koliko će iznositi uglovi između vektora $\mathbf{1}_{10} \in \mathbb{R}^{10}$ i vektora kanonske baze $\{e_1, e_2, \ldots, e_{10}\}$? Koliko će takvi uglovi iznositi za dimenzije $n = 100$ i $n = 1000$? Šta će se dogoditi ako nastavimo da povećavamo dimenziju prostora?

(3 poena)

In [16]:
```
#Za n=10:
m1=np.ones((10,), dtype=int)
m2=np.zeros((10,), dtype=int)
m2[0]=1
cosinus1 = np.dot(m1,m2)/(np.linalg.norm(m1)*np.linalg.norm(m2))
print(np.degrees(np.arccos(cosinus1)))

#Za n=100:
m1=np.ones((100,), dtype=int)
m3=np.zeros((100,), dtype=int)
m3[0]=1
cosinus2 = np.dot(m1,m3)/(np.linalg.norm(m1)*np.linalg.norm(m3))
print(np.degrees(np.arccos(cosinus2)))

#Za n=1000:
m1=np.ones((1000,), dtype=int)
m4=np.zeros((1000,), dtype=int)
m4[0]=1
cosinus3 = np.dot(m1,m4)/(np.linalg.norm(m1)*np.linalg.norm(m4))
print(np.degrees(np.arccos(cosinus3)))

#Priblizavace se 90 stepeni
```

```
71.56505117707799
84.26082952273322
88.18784625299602
```

d) Koliko ima različitih vektora prostora $\mathbb{R}^n$ čije komponente predstavljaju neki niz brojeva iz skupa $\{-1, 1\}$. Na primer, za $n = 2$ to su vektori:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

Za $n = 3$ to su vektori:

$$\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\-1\\1\end{bmatrix}, \begin{bmatrix}-1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\1\end{bmatrix} \begin{bmatrix}-1\\-1\\-1\end{bmatrix}.$$

Kakve će uglove zauzimati ovi vektori sa kanonskom bazom za velike dimenzije n?

(3 poena)

In [18]:
```
m1=np.array([-1,1])
m2=np.array([1,-1])
m3=np.array([-1,-1])
m4=np.array([-1,1])
m5=np.array([-1,1])
cosinus2=np.dot(m1,m2)/(np.linalg.norm(m1)*np.linalg.norm(m2))
cosinus3=np.dot(m1,m3)/(np.linalg.norm(m1)*np.linalg.norm(m3))
cosinus4=np.dot(m1,m4)/(np.linalg.norm(m1)*np.linalg.norm(m4))
cosinus5=np.dot(m1,m5)/(np.linalg.norm(m1)*np.linalg.norm(m5))
print(cosinus2,cosinus3,cosinus4,cosinus5)
```

```
-0.9999999999999998 0.0 0.9999999999999998 0.9999999999999998
```

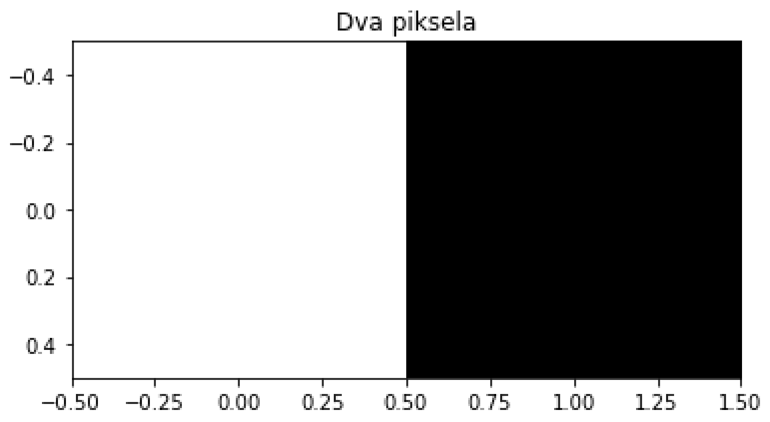
Norma vektora je $\sqrt{n}$,samo treba voditi racuna o znaku skalarnog proizvoda. Iz toga sledi:

$$\cos\alpha = \pm\frac{1}{\sqrt{n}}.$$

Za velike dimenzije, $\alpha \to \pi/2$.

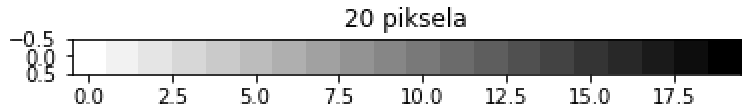
Zadatak 5. Pretpostavimo da su date dve grayscale vrednosti pixela. Aktivirati kodnu ćeliju za njihovo kreiranje. Izvršiti adekvatnu zamenu imena modula u pozivima naredbi.

In [19]:
```
X = np.random.rand(1, 2)
plt.imshow(X, cmap="gray")
plt.title("Dva piksela");
```



Za zumiranje prostora između dva piksela potrebno je da se kreira niz novih piksela koji će povezati prvobitna dva. To može da se uradi ubacivanjem piksela sa vrednostima nijansi između početne dve vrednosti. Aktivirati kod da biste videli primer za postupak zumiranja na dužinu od 20 piksela.

In [20]:
```
w=np.linspace(0,1,20)
zoomX=X[0,0]*(1-w)+X[0,1]*w
plt.imshow([zoomX], cmap="gray")
plt.title("20 piksela");
```

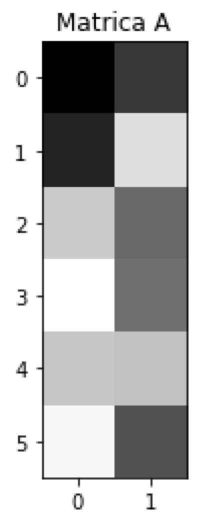


a) Po ugledu na prethodni primer kreirati matricu A piksela (slučajnih vrednosti iz opsega [0,1]) dimenzije 6×2.

In [21]:
```
A=rndm.uniform(0, 1, (6, 2))
```

- **Prikazati dobijenu matricu naredbom matplotlib.pyplot.imshow(A, cmap="gray")**

In [22]:
```
plt.imshow(A, cmap="gray")
plt.title("Matrica A");
```



- **Napraviti zumiranje slike nijansiranjem između kolona ove matrice, na dužinu od 40 piksela. Rezultat smestiti u matricu $zoomA$ dimenzije 6×40.**

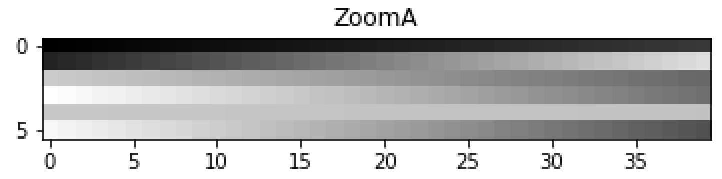
In [35]:
```
a=np.linspace(0,1,40)
a=np.reshape(a,(1,40))
b=np.reshape(A[:,0],(6,1))
c=np.reshape(A[:,1],(6,1))
ZoomA=b@(1-a)+c@a
```

- **Prikazati rezultat naredbom matplotlib.pyplot.imshow(zoomA, cmap="gray")**

(4poena)

In [36]:
```
plt.title("ZoomA")
plt.imshow(ZoomA, cmap="gray")
```

Out[36]:
```
<matplotlib.image.AxesImage at 0x1fce9be7df0>
```

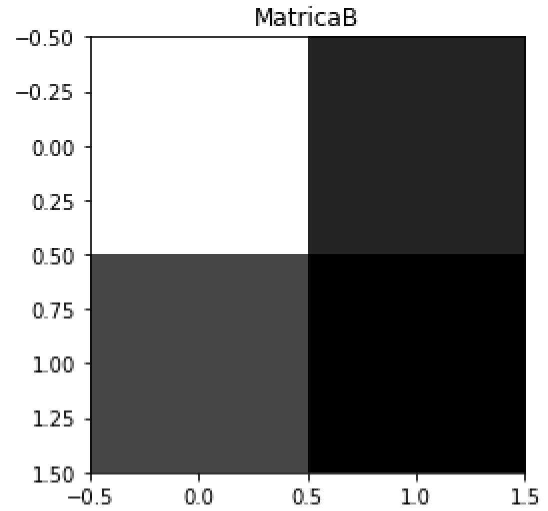


b) Kreirati matricu B piksela (slučajnih vrednosti iz opsega [0,1]) dimenzije 2×2.

In [37]:
```
B=rndm.rand(2,2)
```

- **Prikazati dobijenu matricu naredbom matplotlib.pyplot.imshow(B, cmap="gray")**

In [38]:
```
plt.imshow(B,cmap="gray")
plt.title("MatricaB");
```



- **Napraviti zumiranje slike nijansiranjem između kolona i vrsta ove matrice, na dužinu od 30 piksela i po širini i po dužini. Rezultat smestiti u matricu $zoomB$ dimenzije 30×30.**

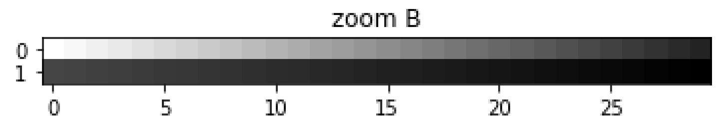
In [44]:
```
a=np.linspace(0,1,30)
a = np.reshape(a,(1,30))
b = np.reshape(B[:,0],(2,1))
c= np.reshape(B[:,1],(2,1))
zoomB=b@(1-a)+c@a
```

- **Prikazati rezultat naredbom matplotlib.pyplot.imshow(zoomB, cmap="gray")**

(6poena)

In [45]:
```
plt.imshow(zoomB, cmap="gray")
plt.title("zoom B")
```

Out[45]: Text(0.5, 1.0, 'zoom B')



Zadatak 6. Neka je matrica $B \in \mathcal{M}_{4\times 4}$. Na B primenjujemo sledeće transformacije:

1. **udvostručimo drugu kolonu;**
2. **prepolovimo treću vrstu;**
3. **dodamo drugoj vrsti prvu;**
4. **zamenimo mesta prvoj i četvrtoj koloni;**
5. **oduzmemo prvu vrstu od svih ostalih;**
6. **zamenimo četvrtu kolonu trećom;**
7. **obrišemo prvu kolonu tako da se broj kolona smanji za 1.**

In [46]:
```
B=np.array([[3,1,7,2], [8,9,18,16],[5,2,8,21],[6,2,13,15]])
print (B)
```

```
[[ 3  1  7  2]
 [ 8  9 18 16]
 [ 5  2  8 21]
 [ 6  2 13 15]]
```

a) Odrediti matricu A koja će izvršiti odgovarajuće transformacije vrsta matrice B. (5 poena)

In [47]:
```
A=np.copy(B)
```

In [56]:
```
#2.prepolovimo trecu vrstu
A[2,:]=A[2,:]/2
#3.dodamo drugoj vrsti prvu
A[1,:]=A[1,:]+A[0,:]
#5.oduzmemo prvu vrstu od svih ostalih
A[:,:]=A[:,:]-A[0, :]
print (A)
```

```
[[ 0  0  0  0]
 [ 8  9 18 16]
 [ 0  0 -1  4]
 [ 3  1  6 13]]
```

b) Odrediti matricu C koja će izvršiti odgovarajuće transformacije kolona matrice B. (5 poena)

In [59]:
```
B=np.array([[3,1,7,2], [8,9,18,16],[5,2,8,21],[6,2,13,15]])
C=np.copy(B)
```

In [60]:
```
#1.udvostrucimo drugu kolonu
C[:,1]=C[:,1]*2
#4.zamena mesta prvoj i cetvrtoj koloni
C[:, [0,3]]=C[:, [3,0]]
#6.zamenimo cetvrtu kolonu trecom
C[:,[2,3]]=C[:,[3,2]]
#7.obrisemo prvu kolonu tako da se broj kolona smanji za jedan
C=C[:,1:]
print (C)
```

```
[[ 2  3  7]
 [18  8 18]
 [ 4  5  8]
 [ 4  6 13]]
```