

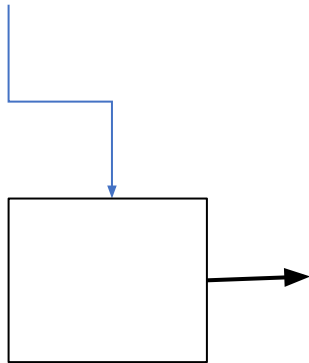
תור

Queue

תור

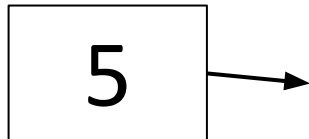
- FiFO – First in First out – מה שנכנס ראשון יוצא ראשון
- void enqueue(Queue q,int x) – Insert x to end of line

enqueue(Queue q,5)



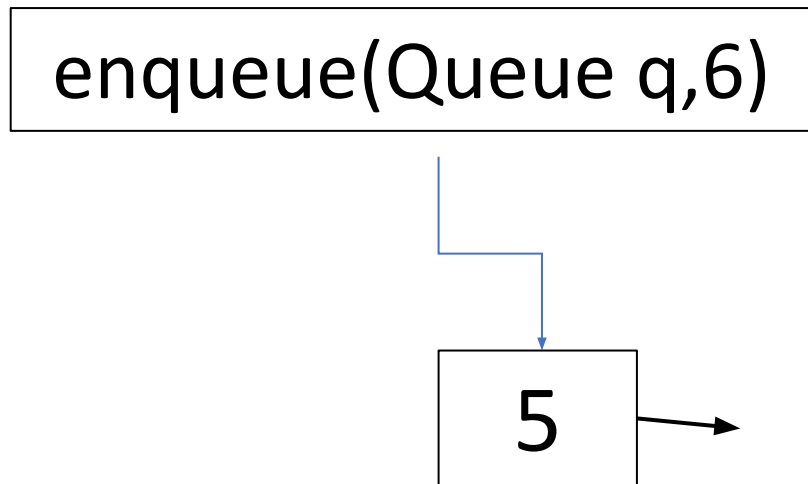
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון



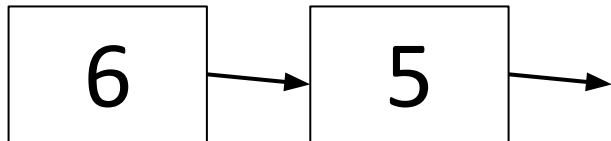
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון



תור

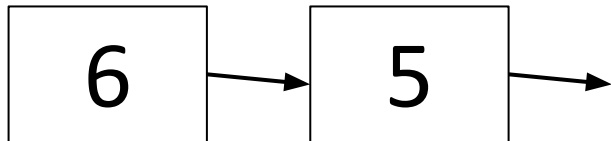
- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון



תור

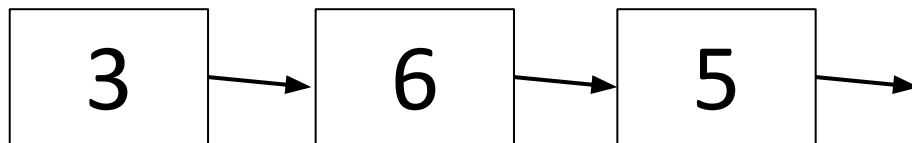
- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון

```
enqueue(Queue q,3)
```



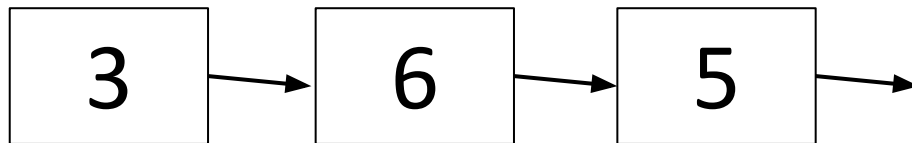
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון



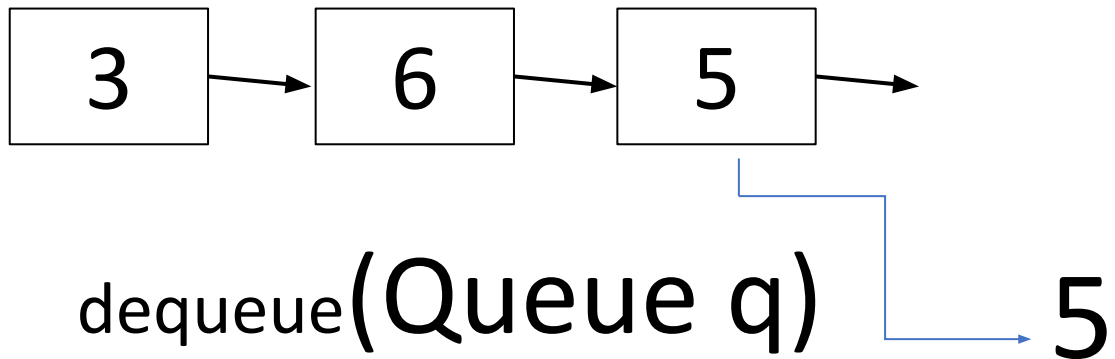
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון
- `int dequeue(Queue q)` – להוציא מספר מראש התור



תור

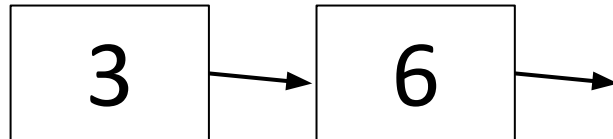
- מה שנכנס ראשון יוצא ראשון – FiFo – First in First out



תור

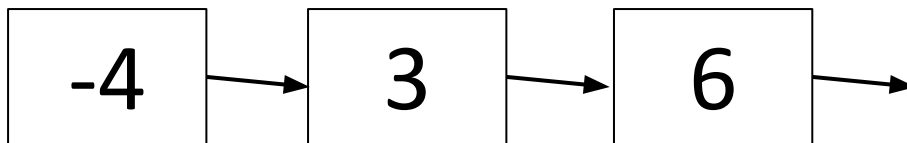
- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון

```
enqueue(Queue q,-4)
```



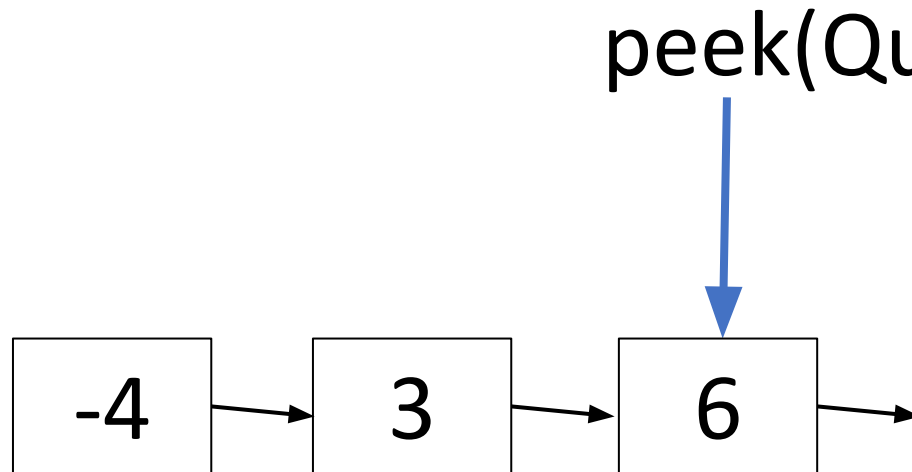
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון



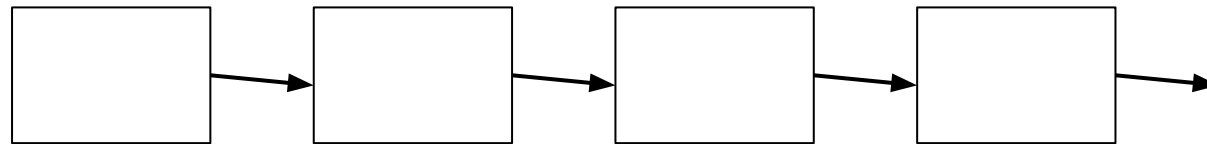
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון
- `int peek(Queue q)` – (הצצה לראש התור (מבלי להוציא את האיבר



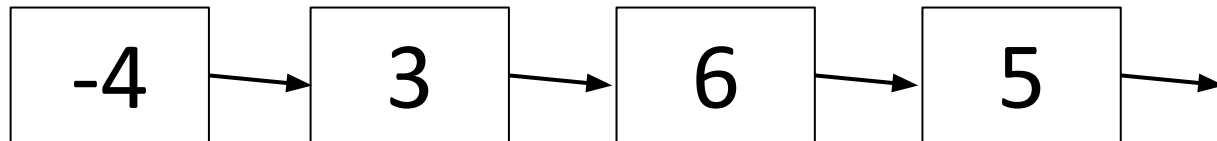
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון
- `int isEmpty()` – האם התור ריק?



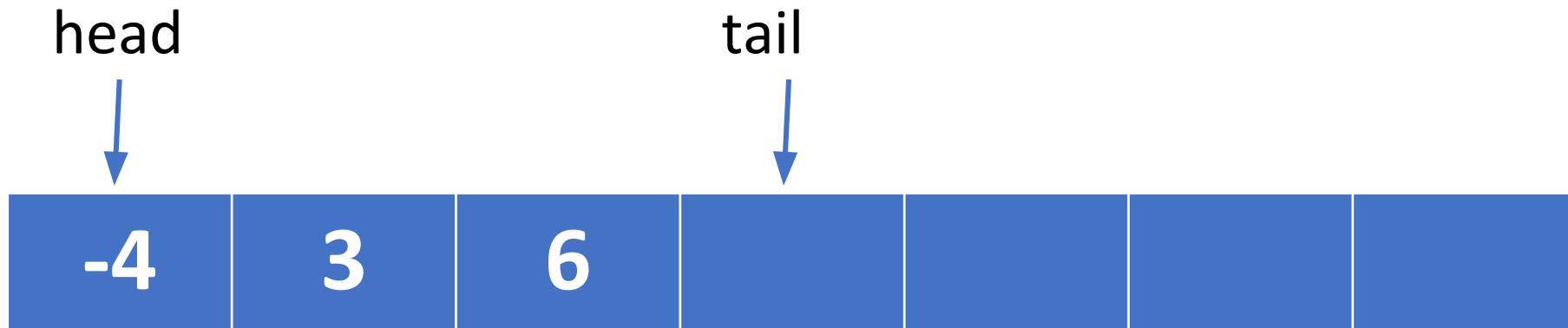
תור

- FiFo – First in First out – מה שנכנס ראשון יוצא ראשון
- `int isFull()` – האם התור מלא



תור באמצעות מערך

- משתנה שיסמן את ראש התור (head) ואת סוף התור (tail)
- head – האינדקס של האיבר הבא לצאת מהתור
- tail – האינדקס הבא אחרי האיבר האחרון שהוכנס (האיבר החדש ייכנס כאן)



תור באמצעות מערך

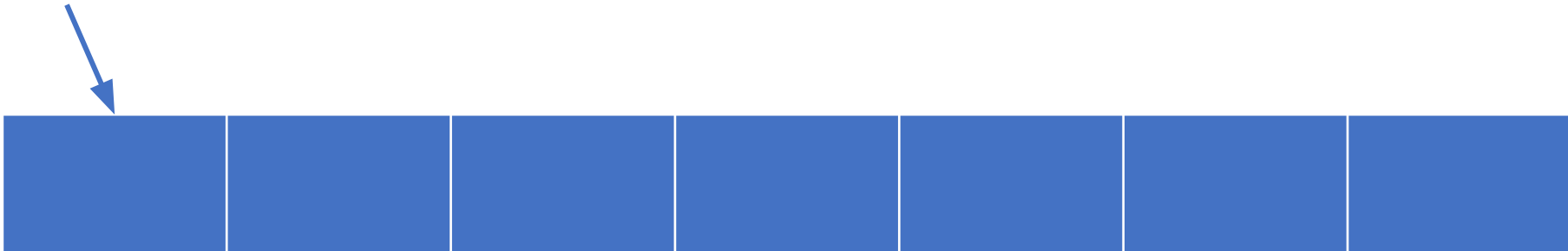
- אתחול

- `;queueSize = 7`

- `;head = tail = 0`

- התור ריק אם `head == tail`

`head = tail = 0`



תור באמצעות מערך

- enqueue(Queue q, int num)
 - queue[tail] = num;
 - tail = tail + 1;



תור באמצעות מערך

- enqueue(Queue q, num)
 - queue[tail] = num;
 - tail = tail + 1; ???



תור באמצעות מערך

- enqueue(Queue q, num)
 - queue[tail] = num;
 - If (tail == queueSize)
 - tail = 0;
 - else
 - tail = tail + 1;



תור באמצעות מערך

- התור מסומן כריק למרות שבפועל הוא מלא.

- $\text{head} == \text{tail}$ – התור ריק



תור באמצעות מערך

- נסמן תור מלא אם $\text{head} = \text{tail} + 1$ בצורה מעגלית
- $\text{head} == (\text{tail} + 1) \% \text{queueSize}$
- תמיד יהיה תא אחד ריק
- לפני הכנסה enqueue תמיד צריך לבדוק האם התור מלא או לא.



תור באמצעות מערך

- enqueue(Queue q, int num)

- if (isFull(q))
 - Error
- queue[tail] = num;
- If (tail == queueSize-1)
 - tail = 0
- else
 - tail = tail + 1;

- isFull(Queue q)

- if (head == (tail+1)%queueSize)
 - return 1;
- return 0;

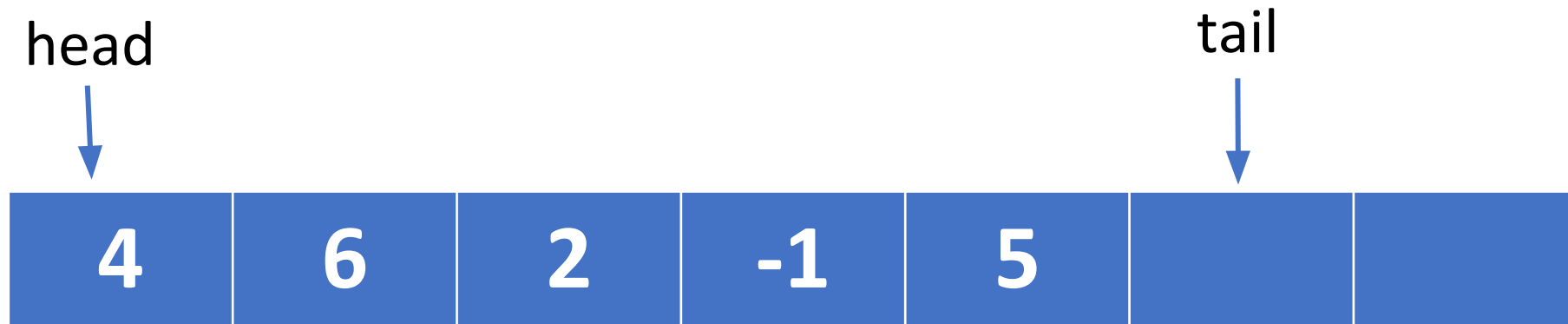


תור באמצעות מערך

- `dequeue(Queue q)`

- `x = queue[head];`
- `head = head + 1;`
- Return x

x 4

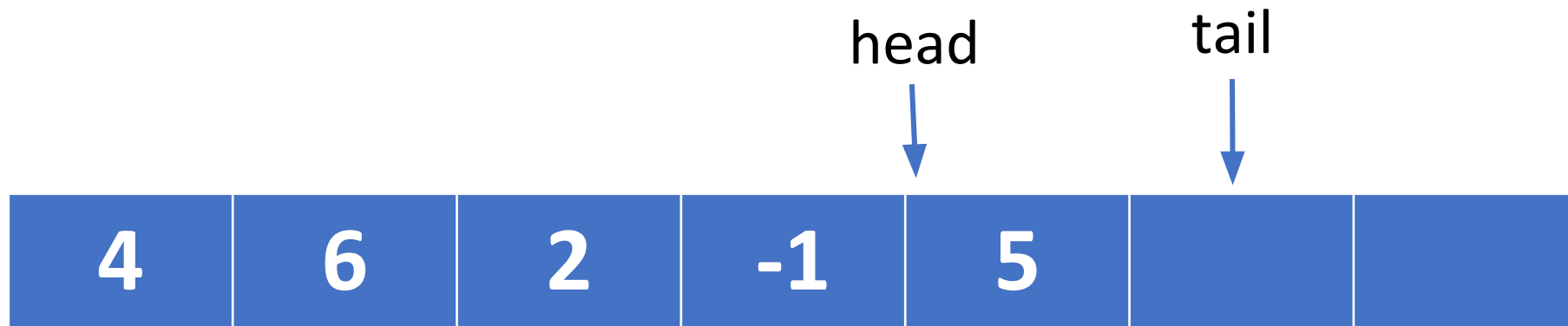


תור באמצעות מערך


- `dequeue(Queue q)`

- `x = queue[head];`
- `head = head + 1;`
- `return x;`

x 5

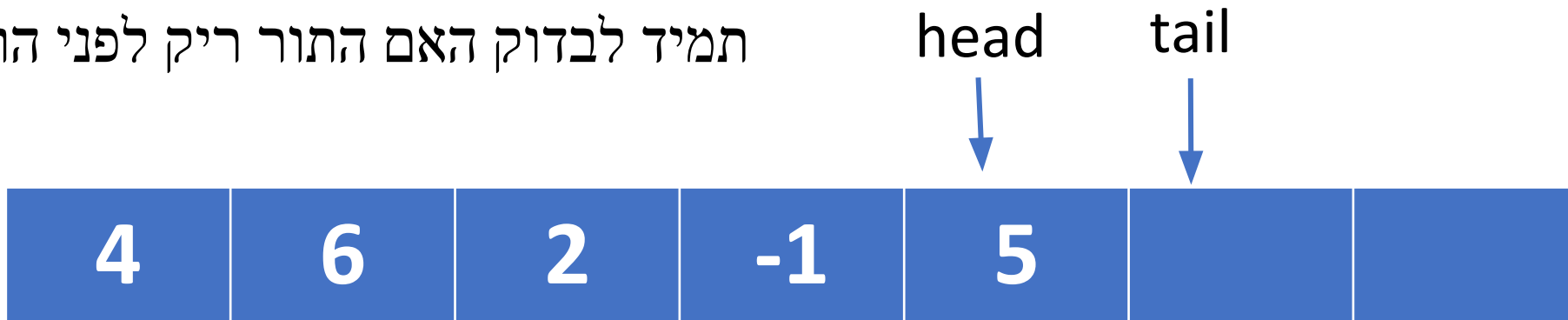


תור באמצעות מערך

- `dequeue(Queue q)` 
 - if (`isEmpty(q)`)
 - Error
 - `x = queue[head];`
 - `head = (head + 1) % queueSize;`
 - `return 1;`

תמיד לבדוק האם התור ריק לפני הוצאה

- `isEmpty(Queue q)`
 - if (`head == tail`)
 - `return 1;`
 - else
 - `return 0;`



תרגיל

- צרו מבנה המכיל מערך בגודל N קבוע של מספרים שלמים, ומשתנים $head, tail$
- כתבו פונקציה `QueueInit` המקבלת את התור ומאתחלת את $head = tail = 0$
- כתבו את הפונקציה `isEmpty` המקבלת פרמטר תור ומחזירה 1 אם הוא ריק, אחרת 0
- כתבו את הפונקציה `peek` המקבלת פרמטר תור ומחזירה את האיבר שבראש התור
- כתבו פונקציה `enqueue` שמקבלת פרמטרים: את התור, ומספר.
הפונקציה תבדוק אם התור מלא, במידה וכן תדפיס שגיאה
במידה והתור לא מלא, תכניס את המספר לתור
- כתבו פונקציה בשם `dequeue` שמקבלת את התור ומוציאה מספר מהתור ומחזירה אותו
- לא לשכוח לעדכן את $head$ ו $tail$ בהכנסה והוצאה, וכן לאתחל את $head, tail$ בהתחלה ל 0

תור באמצעות מחסנית

- מימוש תור ע"י מחסנית:
- תמיד צריך לשמור על האיבר הראשון שהוכנס בראש המחסנית, על מנת שיצא ראשון
- נשתמש בשתי מחסניות.

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

- העבר את כל האיברים ממחסנית A למחסנית B
- הכנס את האיבר החדש למחסנית A
- החזר את כל האיברים מ B ל A

- עכשיו האיבר החדש הוא בסוף המחסנית, וייצא בסוף

A מחסנית

B מתסנית

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחזש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: 5

מחסנית A

מחסנית B

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחזש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: 5

מחסנית A
5

מחסנית B

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחזש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: 1

מחסנית A
5

מחסנית B

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: 1

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר החדש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A

מחסנית B
5

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: 1

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר החדש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A
1

מחסנית B
5

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחזש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: 1

מחסנית A
5
1

מחסנית B

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: 2-

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר המחדש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A
5
1

מחסנית B

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: 2-

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר החדש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A
1

מחסנית B
5

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחזש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: 2-

מחסנית A

מחסנית B
1
5

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: -2

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר המחזש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A
-2

מחסנית B
1
5

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הכנסת איבר חדש:

• העבר את כל האיבר ממחסנית A למחסנית B

• הכנס את האיבר המחודש למחסנית A

• החזר את כל האיברים מ B ל A

הכנס: -2

מחסנית A
1
-2

מחסנית B
5

תור באמצעות מחסנית

- מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר
- הכנסת איבר חדש:

הכנס: -2

- העבר את כל האיבר ממחסנית A למחסנית B
- הכנס את האיבר החדש למחסנית A
- החזר את כל האיברים מ B ל A

מחסנית A
5
1
-2

מחסנית B

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הוצאת איבר:

• להוציא איבר ממחסנית A

מחסנית A
5
1
-2

מחסנית B

תור באמצעות מחסנית

• מחסנית A תהיה המלאה תמיד, מחסנית B היא מחסנית עזר

• הוצאת איבר:

• להוציא איבר ממחסנית A

מחסנית A
5
1
-2

מחסנית B

תרגיל

- צרו מבנה בשם QueueStack המכיל מחסנית אחת שהגדרתם בתרגיל של מחסנית.
- כתבו פונקציה QueueInit המקבלת פרמטר QueueStack ומאתחלת את המחסנית של התור
- כתבו את הפונקציה enqueue המקבלת פרמטר QueueStack ומספר שלם ומכניסה את המספר לראש התור
- כתבו את הפונקציה dequeue המקבלת פרמטר QueueStack, מוציאה את המספר בראש התור ומחזירה אותו.
- כמובן שיש לממש גם את הפונקציות isEmpty ו isFull (בעזרת המחסנית)

תרגיל

- נתון תור הממומש באמצעות מערך
- הפוך את סדר התור, מבלי לגשת ישירות למערך. כלומר, האיבר שכרגע ראשון יהיה אחרון.
- למשל: אם התור הוא $[1,2,3,4]$ (4 נכנס ראשון, אחריו 3, 2 ואז 1) התור החדש יהיה $[4,3,2,1]$
- רמז: השתמשו במחסנית

תרגיל

הסבר במילים כיצד ניתן לממש מחסנית באמצעות תור (ייתכנו כמה תורים)
הראה כיצד נממש את הפעולות $\text{push}(x)$ ו $\text{pop}()$ של המחסנית, ע"י סדרת פעולות
 $\text{enqueue}(\text{Queue } q, x)$ ו $\text{dequeue}(\text{Queue } q)$.

תרגיל

א. כתוב פונקציה המקבלת תור של מספרים שלמים ופרמטר נוסף מספר שלם m .
על הפונקציה להדפיס את m האיברים האמצעיים של התור.
בסוף התהליך התור יישאר ללא שינוי.

לדוגמא, עבור התור $[3, 1, 9, 6, 5, 2]$ והמספר $m=3$ יודפס: $[9, 6, 5]$.

אופן חישוב האיברים האמצעיים: אורך התור (6) פחות 3 (m) חלקי 2 (חלוקה בשלמים): $3/2 = 1$. האיבר האמצעי הראשון יהיה אחרי האיבר 1.

עבור תור בגודל 7 ו $m=3$, $(7-3)/2=2$. האיבר הראשון יהיה אחרי איבר 2.

אין להשתמש במערך שבתור או לגשת למערך בצורה ישירה. יש לבצע את התרגיל רק בעזרת הפעולות של התור Enqueue ו Dequeue – מותר להשתמש בתור או מחסנית נוספים כאוות נפשכם.

תרגיל

- נתון תור שמכיל איברים
- באמצעות שימוש בפעולות התור
- `;enqueue(Queue q,int num`
- `;dequeue(Queue q`
- `()isEmpty`

עליכם למיין את התור בסדר עולה, כך שבפעולת `dequeue(Queue q` הבאה, האיבר הראשון שיוצא יהיה הקטן ביותר, וכן הלאה.

ניתן להשתמש במבני עזר נוספים כמו מחסנית ותור

ניתן להניח שכל איבר מופיע פעם אחת בלבד

אין להשתמש במערך

רמז: ניתן במעבר על התור לדעת מי המספר המינימאלי בתור

רמז: אפשר לכתוב פונקציה שמקבלת תור, מוציאה את המספר המינימאלי בתור ומחזירה אותו