**רשימה מקושרת ממוינת- SortedLinkedList**

**רשימה מקושרת ממויינת:** רשימת שנוצרת מהקצאות זכרון של מבנים כאשר בכל מבנה יש מאפיין מסוג פוינטר next)) שמצביע על כתובת המבנה הבא. המיון מתבצע בעת יצירת הרשימה כלומר כל פריט חדש שיכנס יכנס בהתאמה למקום המתאים לו מבחינת המיון.

הפוינטר של המבנה האחרון מצביע על null.

כך נוצרת רשימה מקושרת ממויינת שנבנית מהקצאות זיכרון של מבנים.

* **כל הפונקציות המבנים והמשתנים וההגדרות שהשתמשנו בהם לתרגיל בניית רשימת מקושרת ייעשה בהם שימוש גם פה**.
* **בנוסף לקוד הזה יש להוסיף מספר פונקציות והם**:

**void addItem(int newItem);**

**struct Item\* findHigher(int val);**

**void locateAfterTail(struct Item\* item);**

**void locateBeforeHead(struct Item\* item);**

**void locateBeforeItem(struct Item\* newItem, struct Item\* itemInList);**

**וכך יראה הקוד:**

void addItem(int newItem)

{

    struct Item\* curr;

    curr = (struct Item\*)malloc(sizeof(struct Item));

    curr->val = newItem;

    if (head == NULL)

    {

        head = curr;

        tail = curr;

        curr->next = NULL;

        curr->prev = NULL;

    }

    else

    {

        struct Item\* highItem = findHigher(curr->val);

        if (highItem == NULL)

        {

            locateAfterTail(curr);

        }

        else

        {

            if (highItem == head)

            {

                locateBeforeHead(curr);

            }

            else

            {

                locateBeforeItem(curr, highItem);

            }

        }

    }

}

void locateAfterTail(struct Item\* item)

{

    item->next = NULL;

    item->prev = tail;

    tail->next = item;

    tail = item;

}

void locateBeforeHead(struct Item\* item)

{

    item->next = head;

    item->prev = NULL;

    head->prev = item;

    head = item;

}

void locateBeforeItem(struct Item\* newItem, struct Item\* itemInList)

{

    itemInList->prev->next = newItem;

    newItem->prev = itemInList->prev;

    newItem->next = itemInList;

    itemInList->prev = newItem;

}

struct Item\* findHigher(int val)

{

    struct Item\* ret = NULL;

    struct Item\* curr = head;

    while (curr != NULL)

    {

        if (curr->val > val)

        {

            ret = curr;

            return ret;

        }

        curr = curr->next;

    }

    return ret;

}