**גיליון תשובות- תרגיל 3 – מבני נתונים (10117)**

|  |  |
| --- | --- |
| שם משפחה: משה  שם פרטי : בר  מס' ת.ז. : 206921777  המרצה : יצחק גנות | שם משפחה: גבריאל  שם פרטי : למפרט  מס' ת.ז. : 206215311  המרצה : יצחק גנות |

**שאלה מספר 1** סמן ב- X במשבצת המתאימה את התשובה הנכונה.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ת ש ו ב ה | | | |
| שאלה |  | 1 | 2 | 3 | 4 |
| 1. | א | x |  |  |  |
|  | ב |  | x |  |  |
|  | ג |  |  | x |  |
|  | ד |  |  | x |  |
|  | ה |  | x |  |  |
|  | ו | x |  |  |  |
|  | ז |  | x |  |  |
|  | ח |  |  | x |  |

## שאלה מס' 2

int biggerSumList(Node \*head1, Node \*head2)

{

**A**

int sum1 = sumListAndFindTail(head1);

int sum2 = sumListAndFindTail(head2);

if (sum1 >= sum2)

return 1;

else

return 2;

}

int sumListAndFindTail(Node \*head)

{

int sum = 0;

Node\* pN = head->next;

while (pN != NULL)

{

sum = sum + pN->num;

pN = pN->next;

}

return sum;

}

Node\* findTail(Node \*head)

{

Node\* pN = head->next;

while (pN->next != NULL)

{

pN = pN->next;

}

return pN;

}

Node \*combine\_lists(Node \*head1, Node \*head2)

{

Node\* tail1=findTail(head1);

Node\* tail2=findTail(head2);

switch (biggerSumList(head1, head2))

{

case 1:

tail1->next = head2->next;

tail2->next = head1;

return head1;

break;

case 2:

tail2->next = head1->next;

tail1->next = head2;

return head2;

break;

}

return NULL;

}

Node \*combine\_lists(Node \*head1, Node \*head2)

{

**B**

LIST \*ResultList = (LIST \*)malloc(sizeof(LIST));

ResultList->tail = &ResultList->head;

if (head2->next == NULL && head1->next == NULL)

return &ResultList->head;

if (head1->next == NULL)

ResultList->head.next = head2->next;

if (head2->next == NULL)

ResultList->head.next = head1->next;

Node \*tail1 = head1->next;

Node \*tail2 = head2->next;

int turn = 1;

int flag1 = 0;

int flag2 = 0;

while (flag1 == 0 || flag2 == 0)

{

switch (turn)

{

case 1:

if (tail1 == head1)

{

tail1 = tail1->next;

}

L\_insert(tail1, tail1->num, ResultList);

tail1 = tail1->next;

break;

case -1:

if (tail2 == head2)

{

tail2 = tail2->next;

}

L\_insert(tail2, tail2->num, ResultList);

tail2 = tail2->next;

break;

}

turn = turn \* -1;

if (tail1 == head1)

flag1 = 1;

if (tail2 == head2)

flag2 = 1;

}

return &ResultList->head;

}

Node \* del\_fun(Node \* head, int \* vec)

**C**

{

if(checkArray(vec)==0)

return head;

Node\* pL = head;

int i = 0; int x = 0;

while (pL != NULL)

{

if (vec[i] == x)

{

L\_delete(pL);

x++, i++;

}

else {

x++, pL = pL->next;

}

}

if (head->next == NULL)

return NULL;

return head;

}

int checkArray(int \* vec)

{

if (!checkIfZero(vec) || !checkIfSorted(vec))

return 0;

return 1;

}

int checkIfZero(int \* vec)

{

if (\*vec != 0)

return 0;

return 1;

}

int checkIfSorted(int\* vec)

{

int i = 1;

int checker = \*vec;

while (vec[i] != -1)

{

if (vec[i] < checker) {

return 0;

}

checker = vec[i];

i++;

}

return 1;

}

## שאלה מס' 3

void RangeSearch( tree\_ptr base, itemType u, itemType v)

{

itemType currenValue = base->info;

if (isEmpty(base))

return ;

if (u < currenValue)

RangeSearch(base->left, u, v);

if (u <= currenValue && v >= currenValue)

printf("%d ", base->info);

RangeSearch(base->right, u, v);

}

int isEmpty(tree\_ptr p)

{

if (p == NULL)

return 1;

return 0;

}

## שאלה מס' 4

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 | ; |
| 4 | ; |

**הערה כללית:**

**שימו לב : בעבור השאלות בהן נתבקשתם לכתוב תכנית בשפת סי יש להכין קובץ נפרד לכל שאלה בנפרד. כל שאלה כזו פרויקט נפרד.**