

ש N101049116  
מספר סידורי: 12  
ת.ז: 308561745

87 מספר תעודת זהות (9 ספרות)  
20465 מספר הקורס  
308561745 מועד

ציון בחינה: 97.  
1- נקודות בשאלה 1.א.  
2- נקודות בשאלה 4.ב.

לשימוש הבודק

חסר בחסדו!  
לפי הפולק  
נכון יס  
אולימפ  
כפר קרע  
קריצת

1.א. חלק לא נכון.  
משתנה שבתוך פונקציה מוגדר  
אולימפ? - סלט וכאמור ניתן לבצע  
int  
(ללא ארמור) אך לא ניתן להשתמש במשתנה  
הוא, מבני ~~לא ניתן לבצע~~ עקב סינון מבני  
שלא יוקצו מהו ערכו (ערך נבדל).

1.ב. לא קבצים נכון ולכרימים לא נכון,  
גלוי אויבה סוף בעולם מבצעים, עמלתי חיסור  
פוינטרים הוא חוקי, אך לאי זמק 831  
חיבור ביניהם לא ולכן גם בעולם הכל לא שכן  
כבולה של פוינטר במספר 'א' הנה מוצאת החיבור  
של הפוינטר גם זכאי 'א' בעמלתי

1.ג. חלק לא נכון, ניתן לקרוא לפונקציות מוגדרות  
מחזק בתכנית הראשית



3. א. הקלע שאו', בשורה 8 (הזולתה הפנימית)  
 דורגת את הערך שקובץ קוצם לכן ב ו  
 בנוסף בעלת השמות ב ו במקום ז יסכס  
 תמיד כך האומר ~~ב~~ העיסון שכן תמיד  
 5. J.

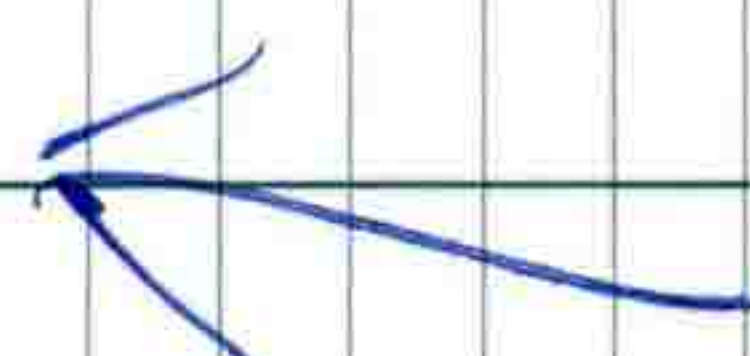
~~א~~ על מנת לתקן נחליף את שורה 8  
 בשורה הבאה:

~~for (j=0; j<3; j++)~~

3. ב. העכירת העכוב כמבוסה, תחילה נעלה  
 לעונקציה print-triangle כערך 4.

בתוך הכונקציה נבדק האם ו < x אם כן  
 מתבצעת קריאה רקורסיבית לעונקציה עם כערך  
 x-1.

לאחר מכן מתבצעת זולתה for שמפסיק  
 את היתו באבית x בעמית ולומר מכן יורדת  
 שורה.





```
#include <stdlib.h>
int main() {
    int *p;
    p = (int *) malloc(10 * sizeof(int));
    return 0;
}
```



2.1c

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define sum_mac(Func_name, type) \
```

```
void * Func_name(void *arr, int size) \
```

```
int i;
```

```
type *sum = (type *) malloc(sizeof(type)); \
```

```
if (sum == NULL) \
```

```
printf("not enough memory"); \
```

```
exit(1); \
```

```
{ \
```

```
*sum = 0; \
```

```
for (i = 0; i < size; i++) \
```

```
*sum = (*sum) + ((type *)arr)[i]; \
```

```
{ \
```

```
return sum; \
```

```
}
```

```
sum_mac(int_sum, int);
```

```
sum_mac(double_sum, double);
```

קוד מקור  
קוד מקור  
ניתן כחוק לפרטים  
ס' 1010 ס' 1010 ס' 1010



```
void *summary(void *arr, int size,
              void *(*func)(void *, int))
{
    if (func == NULL)
        return NULL;
    return (*func)(arr, size);
}
```

7

מספר: 'בני' מקרה שמקרה של פונקציה  
 וסוף. אבנה פונקציה בשם שהתקבל.  
 פונקציה זו מתחילה בוילר מסוף לא יקום (מספר)  
 המכיל את סכום אלכרי המצור, בהמשך אלפים שיתקבל.  
 לאחר מכן תבוא דוגמה לשמור באוקדו.

הפונקציה summary מקבלת מצור, שובל,  
 ובוילר פונקציה ~~שמקבלת~~ void \*  
~~המקבלת~~ void \* מסוף, int, ומתחילה void \*  
 (הבוילר 'ה' פונקציה פונקציה מתקבל).  
 לאחר מחר כתובת 0 במידה !-func הכי  
 את הכתובת 0. (NULL)  
 מומלץ שחרר את ה'צורן של הבוילר  
 בסוף השמור.



2. כח אפניה כי כיוון כיטוב כי סוף סוף כי,

כיוון include, קריאה של הקוד.

```
#define Arr_size 5
```

```
int main() {
```

```
int arr[] = {1, 2, 3, 4, 5};
```

```
int *sum = summary(arr, Arr_size, int_sum);
```

```
if (sum != NULL) { ← כי חזק
```

```
printf("the sum is %d", *sum);
```

```
free
```

```
free(sum); ← כי חזק. כי חזק.
```

```
}  
return 0;
```

```
}  
}
```



```
typedef struct Photographers {
    int id; ← מספר זיהוי
    int days; ← מספר ימי איסוף מיקומים
    int int recc; ← ממוקד המלצה, מספר
    struct Photographers *next, *prev;
} Photographers;
```

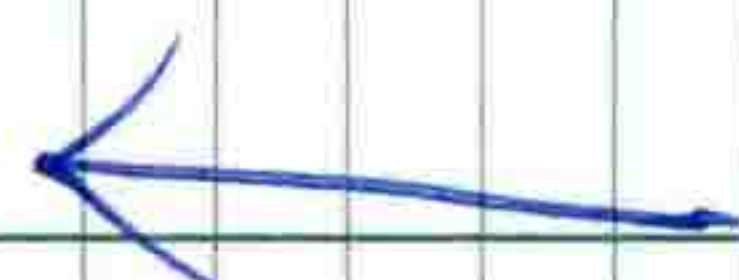
מחברת/מחבר

```
#include <stdlib.h>
#include <stdio.h>

Photographers *Palloc() {
    return (Photographers *) malloc(
        sizeof(Photographers));
}
```

הכונן קצב אצבע הקצאה זכרון.

```
void insert (**head, int id, int days, int recc) {
    Photographers *node = Palloc();
    if (node == NULL) {
        printf("error");
        exit(1);
    }
    node->id = id;
    node->days = days;
    node->recc = recc;
```





১৫০০

```
if(head == NULL) {
```

$(*head) = node;$

$$(*head) \neq prev = (*head);$$

```
(*head) → next = (*head);
```

7 else 5

```
(*head) → prev → next = node;
```

```
node->next = (*head);
```

node  $\rightarrow$  prev = (\*head)  $\rightarrow$  prev;

```
(*head) → prev = node;
```

7

סומך  
הכונן 3'

הַבּוֹנֵה צִדִּיק

$(\text{null } A)$  וואו  $A$  איז א מייטריקס, דער נול וועקטאר.

אזכרה של חברים

\* ביצה : מאפה אביא מצות י"ב ו"ן צורח

NULL	הוא	הוא	הוא	הוא	הוא	הוא
------	-----	-----	-----	-----	-----	-----

~~31/12/20~~





Photographers \*head,  
~~void~~ search(int id)

Photographers \*curr = head;  
 if(head == NULL)

~~printf("the list is empty");~~  
 return NULL;

}

do

if(curr->id == id)

return curr;

curr = curr->next;

while(curr != head);

~~הפונקציה מחפשת בסדרה רשומה בעלת id.~~

~~אלו~~

return NULL;

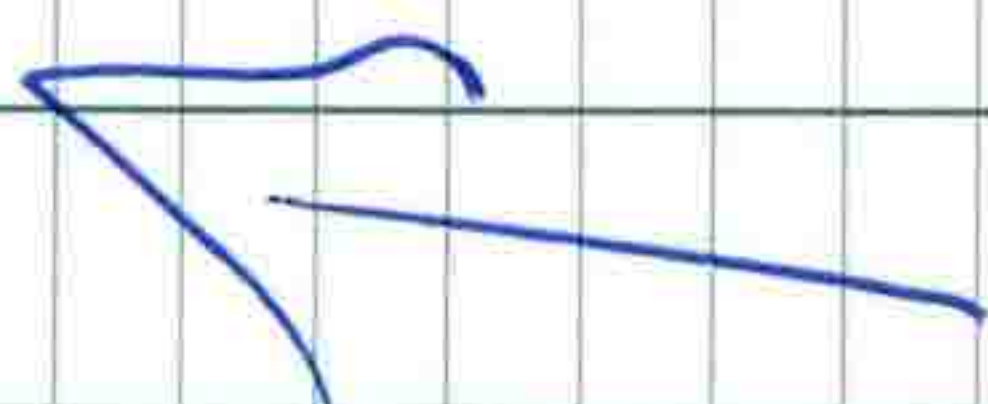
}

הפונקציה מחפשת רשומה בעלת id, ומחזרת

צלף ברצף אולם מספר מניה. במידה ולא קיים

קוצר NULL, מחזרת יחזור בולט se אולם לא.

~~void remove(\*\*head)~~





הנה פונקציה

```
int remove(Photographers **head, int id)
```

```
Photographers *node = search(head, id);
```

```
if (node == NULL) {  
    return 0;
```

הא' ק'ר  
לא נמצא.

}

```
if (node == (*head) && (*head) -> next == (*head)) {  
    (*head)
```

הייק  
כ' יא'ק'ר  
סוף  
כ' יא'ק'ר  
וא'ק'ר  
סוף  
כ' יא'ק'ר  
סוף

```
    (*head) = NULL;
```

```
} else {
```

```
    node -> next -> prev = node -> prev;
```

```
    node -> prev -> next = node -> next;
```

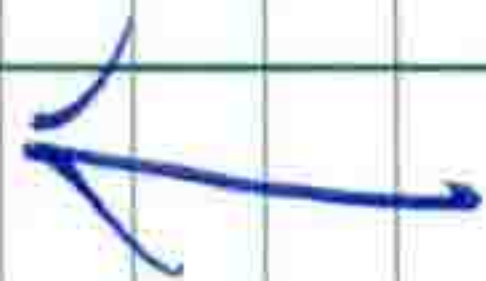
}

```
    return 1;
```

}

הפונקציה מקבלת כ' יא'ק'ר, למטה ומוסרת למטה

כ' יא'ק'ר מוסרת למטה, אם הפונקציה כ' יא'ק'ר יוחס  
1, אחרת, 0.





~~int~~ main()

#define wanna\_remove\_id 5

int main() {

int id, days, recc;

photographers \*head = NULL;

~~while (scanf("%d %d %d", &id, &days, &recc) != 3) {~~

~~insert(&head, id~~

while (scanf("%d %d %d", &id, &days, &recc) != 3) {

~~insert(&head~~

insert(&head, id, days, recc);

}

if (!remove(&head, wanna\_remove\_id))

printf("there is no photographer"

"with the id %d", wanna\_remove\_id);

}

return 0;

התוכנית קוראת 3 מספרים בלבד, ומכניסה

את הנתונים לרשימה, בהתאם, הנתונים יכנסו

לרשימה (כאשר לא ירנו קודם) חוקי (3 מספרים)

הנתונים (מספרים), עומד מכן נרשם קודם לכן, ולכן

הקבוצה היא כללית, כלומר, כללית.