

ש N100611208

מספר
סידורי: 10 ת.ז: 305461501

20465

מספר הקורס

305461501

מספר תעודת זהות (9 ספרות)

לשימוש הבודק

שאלה 1

א) נכון.

אם למשל נמצא חילוק בין שני מספרים ממשיים,
 התוצאה תהיה ממשיה וכן גם אם מספרים שלמים.

ב) אפשריים נכון.

נכון כי ערך כוללים את `<stdio.h>` במכונה, כיוון
 שם מוגדר הקבוע `EOF` כ- `int` (שהוא 15).

ג) נכון.

בעצרת הפונקציה `seek` (שהוא 175) ניתן לקרוא/לכתוב
 לקובץ באופן שרירותי ולא באופן סדרתי.


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

```
#define count_5_bits (arr, size, res) { \
```

```
    res = 0; \
```

```
    int i=0, count; \
```

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

```
        count = 0; \
```

```
        unsigned temp = arr[i], mask = 1; \
```

```
        while (temp) { \
```

```
            if (temp & mask) \
```

```
                count++; \
```

```
            temp <del> <del>
```

```
            temp >>= 1; \
```

```
            if (count == 5) \
```

```
                res++; \
```

```
        } \
```

```
    } \
```

```
} \
```

2/10 10¹⁰

שאלה 2 סוף ב' (הנחתי שניתן למדוד ב-main אופן
צורך פשוטקציה נוספת).

```
int main() {
```

```
    int arr1[] = {2, 4, 7, 12, 16};
```

```
    int arr2[] = {3, 8, 14, 17, 23, 25, 30, 3, 12};
```

```
    int countBits1, countBits2;
```

```
    countBits1 = 0;
```

```
    count_5_bits(arr1, 5, countBits1);
```

```
    count_5_bits(arr2, 9, countBits2);
```

```
    if (countBits1 > countBits2)
```

```
        printf("Arr1 has more numbers!");
```

```
    else if (countBits1 < countBits2)
```

```
        printf("Arr2 has more numbers");
```

```
    else
```

```
        printf("Both arrays have the same  
        number of integers with 5  
        turned on bits");
```

```
    return 0;
```

```
}
```


הקטע שדיו.

משורה 16 -> לא ניתן לעשות free(p) כיוון ש-p
לא תוקצה דינמית באמצעות malloc/alloc/realloc.
וכן צריך למחוק את הקריאה ל-free.
לא ~~צריך~~ שיהיה אלמנט צריך להוסיף כוונת אלמנט אחד.
מספיק, כדי שיהיה אפשר להעביר בין ההעברות (שורה 13).

הקטע שדיו.

משורה 17 -> הוסף לו נכון. ~~צריך~~
if ((s->flag_num_courses &&
(s->flag_average || s->flag_previous)))

צריך להחליף את ההנאי-5

if (s->flag_num_courses && (s->flag_average ||
s->flag_previous))

#define NOT_FOUND 'n' -> 4

ונשנה את שורת 29-30 כדי שיוצגו רק ההסברים

שלא נמצאה להם נמסרה (הסבר הקא.)

אלא שם ההחלפה משורה 18 נשנה ל-

NOT_FOUND.

הנה
3
! (4)


```
for (i=0; i< SIZE_ARR; i++)
    if (id=find_student(arr_students+i))
        id = find_student(arr_students+i);
    if (id != NOT_FOUND)
        printf("Student %s was found\n", id);
```


היקט גרין.

הנחיה בשורה 3 רצה על האמצעים שהתקבלו משרה
 הפקודה ומטה אלמנטים של P . ושלח כל משה
 ופונקציה bin_val .
 הפונקציה bin_val מוודא שהמספר בטווח הרצוי ומציפה
 את הסדר הנכון של המערך הסופי בשורה 9.

~~typedef struct tile~~

```
typedef struct tile {  
    int type;  
    int height;  
    int width;  
    int depth;  
    int strength;  
    char * supplier;  
    int struct tile *next;  
} Tile;
```


סוף ה פסוק

```
Tile * add_tile (Tile * newTile, Tile * head)
{
```

```
if (head->next == head) {
    head->next = newTile;
    newTile->next = head;
}
```

משה ה' 3/17/21
סוף ה פסוק!

```
Tile * temp = head->next;
head->next = newTile;
newTile->next = temp;
```



```
}
```

```
void print_largest (int size, Tile * head) {
```

```
    int count = 1;
```

```
    Tile * curr = head;
```

```
    do {
```

```
        if ((curr->height * curr->width) == size)
```

```
            printf ("At location %d: \n",
```

```
                type: %d, height %d, width %d, depth %d,
```

```
                strength %d, supplier %s \n",
```

```
                count, curr->type, curr->height, curr->width,
```

```
                curr->depth, curr->strength, curr->supplier);
```

```
            curr = curr->next; count++;
```

```
    } while (curr != head); }
```



```


int
find_largest find_largest (Tile * head find_largest) {
    curr = head;
    Tile * curr = head;
    Tile * largest = head;
    while (curr->next != head) {
        do { if ((curr->height * curr->width) >
                (largest->height * largest->width))
                largest = curr;
            curr = curr->next;
        } while (curr != head);
    }
    return (largest->height * largest->width);
}

```

```

int find_location (Tile * head, Tile * find) {
    int count = 1;
    Tile * curr = head;
    while (curr != find) {
        count++;
        curr = curr->next;
    }
    return count;
}

```



 N/A


```
#include <stdio.h>

#define G 1          "ג"
#define C 2          "קמיק"
#define S 3          "ס"

int main (argc, int argc, char * argv[]) {
```

```
    if (argc == 1) {
        printf("no file name");
        exit(1);
    }
```

```
    Tile * head = NULL; (Tile *) malloc(sizeof(Tile));
    FILE * fp;
    fp = fopen(argv[1], "r");
    int res = 0 0;
```

```
    while (fscanf(fp, "%d %d %d %d %d %d",
```

```
        &res,
```

```
        &res,
```

```
        res = fscanf(fp, "%d %d %d %d %d %d",
```

```
            head->type, head->height, head->width,
            head->depth, head->strength,
            head->supplier);
```

```
    head->next = head;
```

```
    if (res != 1) {
        printf("Error");
        exit(1);
    }
```


המלך של main

```
while (res != EOF) {
```

res

```
Tile *temp = (Tile *) malloc(sizeof(Tile));
```

```
fscanf(fp, "%d %d %d %d %d %s",
```

```
temp->type, temp->height, temp->width,
```

```
temp->depth, temp->strength,
```

```
temp->supplier);
```

```
add_tile(temp, head);
```

```
}
```

int

```
Tile *largest = find_largest(head);
```

```
print_largest(largest, head);
```

```
return 0;
```

: פקטור

G, 100, 200, 10, 3, *stone factory

S, 300, 100, 5, 2, Granit place

C, 30, 10, 4, 1, Rocks.