

מקרא נקודות - מבחן מועד א' תשפ"א

תכנות מערכות א'

להלן הורדת הנקודות על טעויות נפוצות.
אם הקוד לא מתקמפל לפחות חצי מהנק' של השאלה יורדות

שאלה 1 (28 נק')

א. מה עושה הפונקציה הבאה?

a. **מינוס 14**: הפונקציה ממירה אותיות קטנות וגדולות מתחילת ה-abc לסוף ה-abc.

$a <-> z, b <-> y, c <-> x \dots$

b. **מינוס 5**: היא משאירה את כל שאר הסימנים מבלי לגעת בהם.

c. **מינוס 5**: כל אמירה לא נכונה אחרת

ב. אלו בעיות יש ואיך אפשר לתקן אותם?

מינוס 5 על כל אחד מהסעיפים הבאים - בעיה או תיקון:

בעיה 1: הפונקציה מחזירה כתובת של מערך שהוקצה על המחסנית של הפונקציה. כאשר הפונקציה מסתיימת, המערך משתחרר, ואם אחר כך ינסו לגשת אליו, זו תהיה גישה לכתובת לא חוקית.

תיקון: להקצות את המערך על הערימה הדינמית, או לקבל את כתובת המערך כפרמטר.
בעיה 2: הלולאה רצה עד האיבר האחד לפני האחרון ולכן לא מעתיקה את התו '\0' מהמחרוזת str1 ל-str2.

תיקון: נשנה את שורה 3: $\text{for}(\text{int } i=0; i < \text{strlen}(\text{str1})+1; i++)\{$

ג. **מינוס 3**: תגרם שגיאת ריצה, עקב גישה לכתובת לא חוקית, כי ניגש לאינדקס שמאותחל עם ערכי זבל.

ד. **מינוס 3**: לא תגרם שגיאה, כי char הוא סוג של int.

ה. **מינוס 3**: (א) תגרם שגיאת ריצה, עקב חריגה מגבולות המערך. (ב) תגרם שגיאת קומפילציה, כי חסר ';' בסוף הפקודה.

שאלה 2 (46 נק')

א. (18 נק')

כל בדיקה תקינות שחסרה - מינוס 3 נקודות

עבור כל שורה אחרת שחסרה או עם טעות – מינוס 5

```
void* malloc(unsigned int byte_num) {  
    if (byte_num==0) {return null};  
    void* mem_start = sys_malloc(ExtraBytes+ byte_num);  
    if(!mem_start) {return null};  
    unsigned int* size = (unsigned int*)mem_start;  
    *size = byte_num;
```

```

        return mem_start + ExtraBytes;
    }

void free(void* ptr) {
    unsigned int* size = (unsigned int*)ptr - ExtraBytes;
    sys_free(size , *size + ExtraBytes);
}

```

ב. (28 נק')

14 נקודות על ההוספה, 14 נקודות על השחרור

אי הגדרה של static - **מינוס 6**

הגדרה של {} מסביב לכל הפונקציות זו טעות קומפילציה - **מינוס 6**

בדיקות תקינות - לא NULL גם ב next וגם malloc - **מינוס 3**

הקצאה של אפס - **מינוס 2**

פנייה לא לכתובת - **מינוס 2**

טעות לוגית אחרת בפונקציות - **מינוס 6**

בעיה בהגדרת structure - **מינוס 6**

```

#define null 0
typedef struct memdesc_ {
    void* ptr;
    unsigned int size;
    struct memdesc_ *next;
} memdesc;

static memdesc* ExtraBytes_LL = null;

void addNode(memdesc* entry) {
    if(ExtraBytes_LL == null) {
        ExtraBytes_LL = entry;
        return;
    }
    memdesc* curr = ExtraBytes_LL;
    while(curr->next != null) {curr = curr->next;}
    curr->next = entry;
}

void* malloc(unsigned int byte_num) {
    if (byte_num<=0) return null;
}

```

```

void* mem_start = sys_malloc(byte_num);
if(!mem_start) return null;

memdesc* entry = sys_malloc(sizeof(memdesc));
if(!entry) {
    sys_free(mem_start, byte_num);
    return null;
}
entry->ptr = mem_start;
entry->size = byte_num;
entry->next=null;
addNode(entry);
return mem_start;
}

unsigned int removeNode(void* ptr) {
    if (ptr==null || ExtraBytes_LL==null) return 0;
    memdesc* prev;
    memdesc* curr = ExtraBytes_LL;
    while(curr != null) {
        if(curr->ptr == ptr) break;
        prev = curr;
        curr=curr->next;
    }
    unsigned int size = curr->size;
    prev->next = curr->next;
    sys_free(curr,sizeof(memdesc));
    return size;
}

void free(void* ptr) {
    unsigned int size = removeNode(ptr);
    sys_free(ptr, size);
}

```

שאלה 3 (26 נק')

חובה לבדוק שהתוכנית עובדת במקרים הבאים:

- א. מערך בגודל אחד ובגודל של יותר מאחד
- ב. עם / בלי אפסים באברי המערך
- ג. עם goal שלא נמצא במערך ושנמצא במערך
- ד. goal שלילי, אפס וחיובי

יצירת מערך עזר ללא שימוש בהקצאת זיכרון דינמית – **מינוס 6**

מעבר על מערך יותר מפעם אחת – כולל לולאה בתוך לולאה – **מינוס 6**

חלוקה ללא בדיקה אם האיבר הוא אפס – **מינוס 6**

שימוש במודולו ללא בדיקה אם האיבר הוא אפס – **מינוס 6**

השמה של אינדקס מחוץ לגבול – **מינוס 6**

כל מי שעשה רק את סעיף ב', כפי שהובהר לכם במבחן, הנקודות יורדות פעמיים - גם עבור סעיף א' וגם עבור סעיף ב'

א. (16 נק')

שימו לב כי אסור לעבור על המערכים הנתונים (b,a) יותר מפעם אחת

```
int mul_elements(int a[], int b[], int len_a, int len_b, int goal) {
    int* help_arr = (int*)malloc(sizeof(int)*len_a);
    for(int i=0; i<len_a; i++){
        help_arr[i]=a[i];
    }
    for(int j=0; j<len_b; j++){
        b_element=b[j];
        for(int i=0; i<len_a; i++){
            if(help_arr[i]*b_element == goal) {
                free(help_arr);
                return 1;
            }
        }
    }
    free(help_arr);
    return 0;
}
```

ב. (10 נק')

```
int mul_elements(int a[], int b[], int len_a, int len_b, int goal) {
    int i=0, j=len_b-1;
    while(i<len_a && j>=0) {
        int mul = a[i]*b[j];
    }
}
```

```
        if(mul == goal) return 1;
        if(mul > goal) j--;
        else i++;
    }
    return 0;
}
```