סמסטר א' תשע"ד, מועד א'

14/2/2014
משך הבחינה: שעתיים
חומר עזר: שני דפי עזר

בחינה בקורס: פרויקט תוכנה
ד"ר ניר אנדלמן, משה סולאמי

הנחיות כלליות לבחינה:

- בראש העמוד הראשון של טופס המבחן (עמוד זה) יש לציין את מספר תעודת הזהות.
- בבחינה ארבע שאלות (פתוחות ואמריקאיות) בעלות ניקוד משתנה, בסך של 100 נק'.
- את התשובות לשאלות 2-4 יש למלא בטבלה המיועדת לכך (בעמוד זה).
- חובה **לתעד** את התשובה לשאלה 1 (כמובן מותר בעברית). בתשובה יש לכלול את כל ההכרזות הדרושות, אך אין צורך להוסיף לקטעי הקוד פקודות #include.

בהצלחה !

טבלת תשובות לשאלות 2-4

שאלה 2	0 – 6	2	10
שאלה 3	1 7	1 7	15
שאלה 4	2 3 5 10 12	2	10

שאלה מס' 1 (65 נק')

נתון קובץ טקסט המכיל מספר לא ידוע של מלים המופרדות על ידי whitespace (רווח, שורה חדשה וטאב). כתבו תוכנית שקוראת את הקובץ ששמו ניתן כארגומנט ב- command line, ובונה רשימה מקושרת של המלים בתוכו (כל איבר ברשימה מכיל מילה), ממויינות לפי סדר עולה.
לאחר מכן התוכנית משתמשת ברשימה כדי להדפיס את המילים בסדר ממויין, מופרדות על ידי רווחים, ולבסוף התוכנית משחררת את כל המשאבים בהם השתמשה.
הנחיות נוספות:

- אפשר להניח שכל מילה בקובץ היא באורך שאינו עולה על 99 תווים.
- במקרה שגיאה, מותר להשתמש ב-assert מבלי לשחרר את משאבי התוכנית.
- לכל מילה ברשימה המקושרת, נחזיק את כמות הזיכרון הדרושה בדיוק בלי בזבוז.

דוגמא: נניח שהקובץ מכיל:

Houston we have a problem
The Eagle has landed

אזי הפלט יהיה:

Eagle Houston The a has have landed problem we

להזכירכם, ייצוג ASCII של אותיות גדולות קטן מזה של אותיות קטנות.

שאלה מס' 2 (10 נק')

מהו הפלט של קטע הקוד הבא משמאל לימין (הניחו קיום כל ההכרזות הדרושות):

```
#define MAC(a,b) (a < b ? a * b : a - b)

void main(void) {
      int i = 2;
      int j = 4;

      int k = MAC(i,j);
      int l = MAC(j,i);

      i = MAC(k+l, k-l);
      j = MAC(k-l, k+l);

      printf("%d %d\n", i, j);
}
```

א. 4 -6
ב. 4 60
ג. 0 -6
ד. 0 60

i = 0
j = -6

0 -6

שאלה מס' 3 (15 נק')

מהו הפלט של קטע הקוד הבא? רשמו את הפלט בטבלת התשובות, משמאל לימין (הניחו קיום כל ההכרזות הדרושות):

```
void foo(int *p1, int *p2) {
      p1 = p2;
      *p1 = *p2 + 1;
}
void bar(int **p1, int **p2) {
      p1 = p2;
      *p1 = *p2 + 1;
      **p1 = **p2 + 2;
}

void main(void) {
      int n[] = { 1, 2, 3 };
      int m[] = { 4, 5, 6 };
      int *p1 = n;
      int *p2 = m;

      foo(p1, p2);
      bar(&p1, &p2);
      printf("%d %d\n", *p1, *p2);
}
```

1, 7

שאלה מס' 4 (10 נק')

מהו הפלט של קטע הקוד הבא משמאל לימין (הניחו קיום כל ההכרזות הדרושות):

```
int s( int x ) { return x*x; }
int c( int x ) { return x+1; }

int comb( int (*t1x)(int),
          int (*t2x)(int),
          int x ) {
     return t1x(t2x(x));
}

void print_tx( int* arr, int length,
               int (*t1x)(int), int (*t2x)(int) )
{
     int *p;
     for(p = arr; p - arr < length; ++p)
          printf ("%d ", comb(t1x, t2x, *p));
}

int main( void )
{
     int arr[] = { 1, 2, 3, 4 };

     print_tx(arr, 4, c, s);
     return 0;
}
```

א. 4 9 16 25
ב. 2 5 10 17
ג. 1 4 9 16
ד. 5 10 17 26

בהצלחה!

לשימוש המרצה בלבד

טבלה לחישוב ציונים

שאלה	ציון
1	65
2	10
3	15
4	10

```
typedef struct node
{
    char *word;
    struct node* next;
};

typedef struct node* list;

list createNode(char *word);
list createList(FILE *ifp);
void merge(list lst);
void printlist(list lst);
```

```
list createNode(char *word)
{
    // יצירת הצומת
    list newnode = (list)malloc(sizeof
                                (node));
    // בדיקת הקצאה
    assert(newnode != null); // בדיקת הקצאה
    newnode->next = null; // אתחול next
    newnode->word = (char*)
    // אתחול המילה
                calloc(strlen(word)+1,
    assert(newnode->word != null, sizeof(char));
/* אתחול המילה */ strcpy(newnode->word, word);
    return newnode;
}
```

```
void melt (list lst)
{
    if( lst==null)                    // אם הרשימה ריקה
        return;
    melt (lst->next);                 // מחיקת הבא
    free(lst->word);                  // שחרור המילה הנוכחית
    free( lst);                       // שחרור המבנה
}
```

מחיקה — מחיקת create list

```
list create List (FILE *ifp)
{
    char buf[400];
    list *lst=null, *curr=null,
        *temp=null;

    int res=0; cmp=0;
    assert (ifp!=null);
    res =fscanf (buf, "%s", ifp);

    while (res != EOF )
    {
        curr= createnode(buf);
        if (lst==null)
        {
            lst = curr;
        }
        else
        {
            temp=lst;
            if (strcmp(lst->word, curr->word) > 0)
            {
                curr-> next=lst
                lst = curr;
            }
```

מיון

```
void printlist (list lst)
{
    assert (lst != null);   // בדיקה שרשימה לא null
    list temp = lst;        // מצביע המשתנה הזמני
    while (temp != null)
    {
        printf("%s", temp->word);   /* הדפסה של המילה אל המסך */
        temp = temp->next;          /* מעבר למילה הבאה ברשימה */
    }
}

int main (int argc, char *argv[])
{
    list lst = null;
    assert (argc == 2);   /* בדיקה שמספר הפרמטרים נכון */
    FILE *ifp = fopen(argv[1], "r");
        assert (ifp != null);
    lst = CreateList(ifp);   // יצירת הרשימה המקושרת
        assert (lst != null);
    printlist(lst);          // הדפסת הרשימה
    melt(lst);               /* שחרור זיכרון */
    fclose(ifp);             // סגירת הקובץ
    return 1;                // סיום התוכנית
}
```

/* מקבל קובץ ומחזיר רשימה ממוינת */

```
list createList(FILE *ifp)
{
    char buf[100];   // מערך תווים זמני
    list lst=null, temp=null, curr=null;
    int res=0, cmp=0;
    assert(ifp!=null);   // בודקים שלא null
    ~~res=fscanf(buf,"%s",ifp);~~
    /*קריאה*/ res=fscanf(ifp,"%s",buf);
    /*תנאי לולאה*/ while(res!=EOF)
    {
        curr=createNode(buf);   // יצירת תא חדש
        if(lst==null)   // רשימה ריקה
            {lst=curr;}
        else
        {
            temp=lst;   // מעבר לרשימה
            cmp=strcmp(lst->word,curr->word);   // השוואה בין המילה הנוכחית לאיבר הבא
            if(cmp>=0)
            {
                curr->next=lst;
                lst=curr;
            }
```

/* הרשימה ~~[illegible]~~ ממוינת
בסדר עולה
אנחנו מוסיפים איבר שנכון
~~לרשימה~~ לעבור
מהאיבר הראשון
אל
לשים
לעקוף המשתנים
הבא
הישן
*/

```
else
{
        while ( temp->next != null)
        {
/* בדיקה
   האם
   המילה
   הבאה
   גדולה מהנוכחית */
        cmp = strcmp(temp->next->word,
                     curr->word);

        iF( cmp >= 0)
        {
// עדכון הרשימה    curr->next = temp->next;
                   temp->next = curr;
                   break;

        }
/* קידום temp */   temp = temp->next;

        }


        iF( curr->next == null)
        {
// curr לא עודכן   cmp = strcmp ( temp->word,
                                 curr->word);

// curr בסוף מקום   iF ( cmp <= 0)
// עדכון            { temp->next=curr;}
        ~~else~~
        }

}
}
// קליטת המילה הבאה   res= Fscanf (iFp, "~~%s~~ "%s", buF);
}
סוף
```

מבוא לכלכלה 1

מבחן מועד א' 2015

65/65