1045 מ

37

מחברת מס' _____

מתוך _____ מחברות

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

לפני התחלת הבחינה מלא את כל הפרטים הבאים בכתב ברור וקרא בעיון את ההוראות:

1. על הנבחן להיבחן רק בחדר שבו הוא רשום.
2. עם הכניסה לחדר הבחינה יש להניח את החפצים בצד לרבות מכשירי קשר ואמצעי תקשורת אחרים כשהם כבויים.
3. אסור להחזיק בהישג יד, בחדר הבחינה או בסמוך לו, כל חומר הקשור לבחינה/לקורס פרט לחומר שהשימוש בו הותר בכתב על ידי המורה.
4. יש למלא את הפרטים על מחברת הבחינה במקום המיועד לכך בלבד. אין לכתוב את השם או כל פרט מזהה אחר בתוך המחברת.
5. יש להישמע להוראות המשגיח. נבחן לא יעזוב את מקומו ללא קבלת רשות המשגיח. הפונה בשאלה או בבקשה ירים את ידו.
6. נבחן שנכנס לחדר הבחינה וקיבל את השאלון (טופס הבחינה) לידו ייחשב כמי שנבחן במועד זה. היה והחליט לא לכתוב את הבחינה, לא יהא רשאי לעזוב את חדר הבחינה, אלא כעבור חצי שעה ממועד תחילתה ולאחר שהחזיר את המחברת והשאלון. ציונו בבחינה יהיה "0".
7. קריאת השאלון מותרת רק לאחר קבלת רשות המשגיח.
8. יש לכתוב את התשובות בעט, בכתב יד ברור ונקי. נבחן הבוחר לכתוב טיוטה יעשה זאת בעמודו הימני של דפי מחברת הבחינה ויציין בראש העמוד "טיוטה". אין לתלוש דפים מהמחברת.
9. מחברות הבחינה שקיבל הנבחן תהיינה בפיקוחו ובאחריותו במשך כל הבחינה. בעת יציאה מן החדר יופקדו המחברות והשאלון בידי המשגיח.
10. בתום הבחינה יחזיר הנבחן את המחברות והשאלון ויקבל מידי המשגיח את כרטיס הנבחן.
11. הנוהג בניגוד להוראות ול"מהל סדרי בחינות ודיווח ציונים" צפוי להפסקת בחינתו ואף להעמדה לדין משמעתי.
12. אין לכתוב מעבר לקו האדום משני צידי הדף.

בהצלחה.

לשימוש המורה הבוחן:

הציון _____

המחברת נבדקה ביום _____

חתימת המורה _____

תאריך הבחינה 19/6/05

שם הקורס תיכנות 1

שם המורה רודיג שרן

החוג/המגמה מדעי המחשב

79839

5	(5)
5	(6)
20	(1)
5	(4)
5	(7)
25	(2)
40	(3)
105	

סמסטר ב' תשס"ה, מועד א'

19/6/2005
משך הבחינה: 3 שעות
חומר עזר: שני דפי עזר

בחינה בקורס: תוכנה 1
מרצה: ד"ר רודד שרן

הנחיות כלליות לבחינה:

- בראש העמוד הראשון של טופס המבחן (עמוד זה) יש לציין את מספר תעודת הזהות.
- בבחינה שבע שאלות (פתוחות ואמריקאיות) בעלות ניקוד משתנה, בסך של 105 נק'.
- את התשובות לשאלות האמריקאיות יש למלא בטבלה המיועדת לכך (בעמוד זה).
- חובה **לתעד** את התשובות לשאלות הפתוחות (כמובן מותר בעברית).
- בתשובות לשאלות הפתוחות יש לכלול את כל ההכרזות הדרושות, אך אין צורך להוסיף לקטעי הקוד פקודות #include.

בהצלחה !

טבלת תשובות לשאלות האמריקאיות

שאלה 4	ב
שאלה 5	ב
שאלה 6	ב
שאלה 7	ד

שאלה מס' 1 (20 נק')

כתוב/כתבי תכנית המקבלת כארגומנטים ב-command line שתי מחרוזות המייצגות שמות קבצי טקסט: קובץ קלט וקובץ פלט. קובץ הקלט מכיל מספר לא ידוע של שורות, שבכל אחת עד 80 תווים (כולל ה-newline). כל שורה מכילה מלים (אחת או יותר) שרווחים ו/או פסיקים מפרידים ביניהם. על התכנית למיין בסדר לקסיקוגרפי את המלים בכל שורה של קובץ הקלט (מלים זהות יהיו רצופות בסדר זה) ולכתוב את השורה המתקבלת בקובץ הפלט, עם רווח יחיד בין המלים.

שאלה מס' 2 (25 נק')

נגדיר את טיפוס הנתונים הבא, של איבר ברשימה מקושרת המחזיק מחרוזת:

```
typedef struct _StringNode {
  char *str;
  struct _StringNode *next;
} StringNode;
```

עליך לממש את הפונקציה הבאה:

```
char *replace(char *text, StringNode *pattern, StringNode *replacement);
```

הפונקציה מקבלת כקלט מחרוזת טקסט, ושתי רשימות בעלות אורך זהה של מחרוזות. על הפונקציה להחזיר מחרוזת חדשה, המתקבלת ממחרוזת הטקסט ע"י החלפת כל מופע של מחרוזת מהרשימה הראשונה במחרוזת המתאימה מהרשימה השנייה (כלומר, מופע של המחרוזת המופיעה ראשונה ברשימה הראשונה יוחלף במחרוזת המופיעה ראשונה ברשימה השנייה וכו'). ניתן להניח שבמחרוזת הטקסט אין מופעים חופפים של מחרוזות מהרשימה הראשונה, וכן שמחרוזת התוצאה אינה קצרה ממחרוזת הטקסט. שים/שימי לב ששתי הרשימות הנתונות יכולות להכיל את אותה המחרוזת (ראה/י דוגמה).

דוגמה:

text: cat dog catcat dog bird horse bird cat
pattern: cat dog bird
replacement: dog cat horse

result: dog cat dogdog cat horse horse horse dog

שאלה מס' 3 (40 נק')

גרף לא מכוון מתואר ע"י אוסף של קודקודים שכל אחד מחזיק את רשימת שכניו כלהלן:

```
typedef struct node_list {
  struct node *node;
  struct node_list *next;
} NList;

typedef struct node {
  NList *nbrs;
  int id;
} Node;
```

א. (20 נק') כתוב/כתבי פונקציה המקבלת מצביע לקודקוד בגרף, ומחזירה רשימה מקושרת של קודקודים הנמצאים במרחק לכל היותר 2 ממנו בגרף (היינו, הרשימה תכלול את הקודקוד, שכניו ושכני שכניו – ללא כפילויות). הגדרת הפונקציה (prototype) היא:

```
NList *get_2nbrs( Node *root );
```

ב. (20 נק') כתוב/כתבי פונקציה המקבלת מצביע לקודקוד v בגרף ומחרוזת המציינת שם של קובץ קלט. הקובץ מכיל מספרים שלמים בפורמט בינארי. המספר הראשון n הוא חיובי ומציין כמה מספרים <u>נוספים</u> יש בקובץ. המספרים הבאים מציינים מספרי זהות (id) של קודקודים בגרף $u_1,\ldots,u_n$ שהם שכנים של v. על הפונקציה להוריד מהגרף את הקשתות $(v,u_1),\ldots,(v,u_n)$ על ידי הורדת $u_1,\ldots,u_n$ מרשימת השכנות של v, והורדת v מרשימות השכנות של $u_1,\ldots,u_n$. הגדרת הפונקציה היא:

```
void remove_nbrs( Node *v, char *fname );
```

שאלה מס' 4 (5 נק')

נתון קטע הקוד הבא (הנח קיום כל ההכרזות הדרושות):

```
void f(unsigned int *x, int i, int j)
{
	*x = (*x>>i) & ~(~0<<(j-i+1));
}

int main()
{
	unsigned int x = ~8;

	f(&x,1,5);
	printf("%u\n", x);
	return 0;
}
```

מה יהיה פלט התוכנית?

א. 5
ב. 14
ג. 27
ד. 32

שאלה מס' 5 (5 נק')

נתון קטע הקוד הבא (הנח קיום כל ההכרזות הדרושות):

```
int a = 0;
int b = 1;

void func(int a)
{
        static int b=0;

        a++;
        b = b + a;

        printf( "%d ", b);
}

int main()
{
        func (++a);
        func (b);
        printf("\n");
        return 0;
}
```

מה יהיה פלט התוכנית (משמאל לימין)?

א. 1 2
ב. 2 3
ג. 2 4
ד. 3 7

שאלה מס' 6 (5 נק')

נתון קטע הקוד הבא (הנח קיום כל ההכרזות הדרושות):

```
#define mp(A,B) A**B

int main()
{
        int a[2], *b, *c;

        a[0] = 5;
        a[1] = 7;

        c = a;
        (*c)++;
        b = c+1;

        printf("%d\n", mp(a[0],b));
        return 0;
}
```

מה יהיה פלט התוכנית?

א. 35
ב. 40
ג. 42
ד. 49

שאלה מס' 7 (5 נק')

מה מבצע ה-script הבא:

```
#!/bin/csh -f

if ($#argv != 1) then
	echo "Error!"
	exit 1
endif

@ x=0
set lines = `wc -l $1`
@ num = $lines[1]

while ($num > 0)
	set next = `head -$num $1 | tail -1 | wc -c`
	if ($next[1] > $x) then
		@ x = $next[1]
	endif
	@ num--
end

echo $x
```

א. מדפיס את מס' התווים בשורה בעלת מס' התווים הגדול ביותר בקובץ המועבר כפרמטר.
ב. מדפיס את אינדקס השורה הארוכה ביותר בקובץ המועבר כפרמטר.
ג. מדפיס את מס' השורות בקובץ המועבר כפרמטר.
ד. מדפיס את השורה האחרונה בקובץ המועבר כפרמטר.

בהצלחה!

שאלה 1

```
#define MAX_LINE 80
int comp_str (void *s1, void *s2)   /* פונק' השוואה ל qsort */
    { return strcmp(*(char **)s1, *(char **)s2)); }

/* פונק' מקבלת שורת קלט ומקרע למילים ומכניסה אל המערך
int split_to_words (char *line, char **words)
{
    int n_words=0;
    words[0] = strtok(line, " ,");   /* המפרידים ' ' ו- ',' */
    while (words[n_words] != NULL)
        words[++n_words] = strtok(NULL, " ,");
    return n_words;   /* מחזיר כמה מילים היו בשורה */
}

/* פונק' לכתיבת שורה נתונה מתוך מערך ( מחזירה 0 בהצלחה . מחזירה -1 בכישלון )
int write_words (char **words, int n_words, FILE *out_f)
{   int ret,i;
    if (n_words ==0) return 0;
    ret = fprintf("%s", words[0]);   /* נדפיס מילה ראשונה */
    if (ret != strlen(words[0])) return -1;
    for( i=1; i< n_words; i++)
    {
        ret = fprintf(" %s", words[i]);   /* נדפיס רווח ואחריו נוספת */
        if (ret != (strlen(words[i]) +1)) return -1;
    }
    if ( fputc('\n', out_f) == 0 ) return -1;   /* נדפיס שורה */
    return 0;   /* בהצלחה מחזיר 0 */
}

int main (int argc, char **argv)
{
    char line[MAX_LINE+1];   /* מערך לשורת קלט */
    char *words[MAX_LINE+1];   /* מערך זרוק מצביעים למילים */
    int n_words;
    int ret;
    char *c_ret;
```

```
FILE *in_f, *out_f;

in_f = fopen(argv[1], "r");      /* פותח קובץ קריאה */
if (in_f == NULL) return -1;
out_f = fopen(argv[2], "w");     /* פותח קובץ כתיבה */
if (out_f == NULL)
  {
     fclose(in_f);
     return -1;
  }
while ( feof(in_f) != 0 )        /* לא הגענו לסוף הקובץ */
 {
    c_ret = fgets(line, MAX_LINE, in_f);
    if ( c_ret == NULL )         /* בעיית קריאה */
     {
      fclose(in_f); fclose(out_f); return -1;
     }
    n_words = split_to_words (line, words);   /* מפצל למילים */
    qsort(words, n_words, sizeof(char *),     /* מיין ע"י qsort */
                  comp_str);
    ret = write_words (words, n_words, out_f);   /* כותב המילים המאוחדות לקובץ */
    if (ret == -1)
     {
      fclose(in_f); fclose(out_f); return -1;
     }
 }
fclose(in_f);
fclose(out_f);
return 0;
}
```

- התוכנית קוראת שורה, מפצלת למילים בעזרת פונקציה split_to_words. מיינת בעזרת qsort וכותבת לקובץ בעזרת פונקציה write_words.
- הכיתוב נעשה בעזרת strtok שבודק מי הוא תו המקור אך לא לדאוג מפרוט לגבי למילים עדיין.

שאלה 2

```
char *replace(char *text, StringNode *pattern, StringNode *replacement)
{
   int text_len = strlen(text);
   char *res = (char *)malloc(text_len+1);
   int res_len = 0, res_memory = text_len;   /* מה זה שיורד מהם
                                                 ירדון אין כלום */
   int i, j;
   StringNode *pat, *rep;
   if (res == NULL) return NULL;   /* בדיקת אלוקציה ריקה */
   for (i=0; i<text_len; i++)
   {
      for(pat=pattern, rep=replacement; pat != NULL; pat=pat->next,
                                                     rep=rep->next)
      {
         /* בדיקה אם התחלת pat מופיע ברצף */
         if ( strncmp(text+i, pat->str, strlen(pat->str)) == 0)
         {
            if ( (res_memory - res_len) <= strlen(rep->str))
            {                    /* בדיקה עוד כלום ירדון */
               res = realloc(res, res_memory * 2 + 1);
               if (res == NULL) return NULL;
               res_memory *= 2;
            }
            /* מעדכן את התחלת הפלט */
            if(strcpy(res + res_len, rep->str) == NULL)
            { free(res); return NULL; }
            res_len += strlen(rep->str);
            i += (strlen(pat->str) - 1);   /* i גדל ב- 1 בסוף ה- for */
            break;
         }
      }   /* סוף ה- for */
      if (pat == NULL)   /* סימן שלא התאים כלום - הגענו לסוף ה- for */
      {
         if ( (res_memory - res_len) <= 1 )   /* בדיקה עוד ירדון */
         {
```

```
        res = realloc(res, res_memory * 2+1);
        if (res == NULL) return -1;
        res_memory *= 2;
    }
    /*                מעתיק תו בודד מהמקור                */
    res[res_len] = text[i];
    res_len++;
  }  /*  for  סוף  */
  res[res_len] = '\0'   /* סיום המחרוזת */
  return res;
}
```

25/25

הבודק נוכח מאמץ תשובה הוגנת. הוא שם דגש על תנאי הבודקים לא מלה ב- get שאינה. אין כן, מעתיקה אותה (ומגדיל שם מסמן גודלו בגלל res_memory). כמובן מוגבלים לשינויים בהקצאה. אין לנו גם אחריות לשחרור, מעתיקים תו בודד מהקלט ומגדילים הלאה. בסוף מוסיפים '\0' לאחרונה תשובה.

שאלה 3 (א)

/* בודק שמתווסף איבר ל- NList ומחזיר מצביע לשורש */

```
NList *insert(NList *lst, Node *node_n)
{
    NList *new_n, *p;

    if (lst==NULL)
    {
        new_n = (NList *)malloc(sizeof(NList));
        if (new_n == NULL) return NULL;
        new_n->next = NULL;
        new_n->node = node_n;
        return new_n;
    }
    p = lst;
    while (p->next != NULL)   /* בודק אם הוא נמצא node */
    {
        if (p->node->id == node_n->id)
            return lst;
        p = p->next;
    }
    /* אם הגענו לכאן, הוא node לא נמצא, נוסף בסוף */
    new_n = (NList *)malloc(sizeof(NList));
    if (new_n == NULL) return NULL;
    new_n->next = NULL;
    new_n->node = node_n;
    p->next = new_n;
    return lst;
}
```

```
NList *get_2nbrs(Node *root)
{
    NList *nbrs = insert(NULL, root); /* אין root בלבד בגודל רשימה */
    NList *nbr1, *nbr2;

    if (nbrs == NULL) return NULL;
    for(nbr1 = root->nbrs; nbr1 != NULL; nbr1 = nbr1->next)
    {
        /* מוסיף את השכן לרשימה */
        nbrs = insert(nbrs, nbr1->node);
        if(nbrs == NULL) return NULL;
        /* עוברת על השכנים של השכן. */
        for(nbr2 = nbr1->node->nbrs; nbr2 != NULL;
                                    nbr2 = nbr2->next)
        {   /* מוסיף שכן מקביה 2 */
            nbrs = insert(nbrs, nbr2->node);
            if(nbrs == NULL) return NULL;
        }
    }
    return nbrs;
}
```

20/20

* במקרה של כישלון בהקצאת זיכרון בשכן, הפונקציה מחזירה NULL
* insert מכניסה node ומקום לרשימה. היא מחזירה פוינטר לרשימה המעודכנת, למקרה שהיה ריק או בגלל הקצאת זיכרון.
* מכניסים את root לרשימה, עוברים על שכניו ומוסיפים אותו ואת השכנים לרשימה ומחזירים.

שאלה 3 (ב)

```
/* פונקציה שמוחקת node מהמילון NList לפי מספר מזהה */
NList *remove_1(NList *lst, int id)
{
    NList *p = lst;          /* רשימה */
    NList *tmp;              /* משתנה עזר */
    if (lst == NULL) return NULL;
    if (p->node->id == id)   /* בדוק להוריד את הראשון */
    {
        tmp = p;
        p = p->next;
        free(tmp);
        return p;
    }
    while (p->next != NULL)
    {
        if (p->next->node->id == id)   /* בדוק להוריד את הבא */
        {
            tmp = p->next;
            p->next = p->next->next;
            free(tmp);
            return lst;
        }
        p = p->next;          /* מקדם את p */
    }
    return lst;
}
```

```
void remove_nbrs(Node *v, char *fname)
{
    FILE *fp = fopen(fname, "rb");
    int *ids;
    int i,j; ret;
    NList *p;
    unsigned int size;
    if(fp==NULL) return;
    /* קוראים רק את ה id */
    ret= fread(&size, sizeof(unsigned int), 1, fp);
    if (ret==0) return;
    ids = (int *)malloc(sizeof(int) * size);
    if(ids==NULL) {fclose(fp); return;}
    /* קוראים לתוך מערך ע"מ ה id */
    ret= fread(ids, sizeof(int), size, fp);
    if (ret != size)
    {
      free(ids);
      return;
    }

    /* נעבור על כל השכנים של v */
    for(p=v->nbrs; p!=NULL; p=p->next)
    {   /* נבדוק אם השכן מופיע ברשימה שקראנו מהדיסק */
        for(i=0; i<size; i++)
        {
            if( ids[i] == p->node->id )
            {  p->node->nbrs =
               remove1(p->node->nbrs, v->id);
            } break;
        }
    }
}
```

* אחרי שמוחקנו ל-v את כל השכנים שלו שמופיעים ברשימה, נמחוק אותם מהרשימה שלו

```
for(i=0; i<site; i++)
  {
    v->nbrs = remove_1(v->nbrs, ids[i]);
  }


free(ids);
fclose(fp);
}
```

* remove_1 מוחקת node נתון (לפי id) מרשימה מקושרת ומחזירה מצביע לרשימה המעודכנת (סיום מקרים שם הרשימה באורך 1 או פחות מוחקים את הראשון).

* קוראים מהקובץ את השכנים שצריך להוסיף. סורקים שם השכנים וכותבים מי מהם ברשימה. אין שום ברשימה מוחקים ל-v מרשימת השכנים שלו.

* אולי סורקים x שם בלה שצריך להוסיף, ומפעילים את remove_1 על רשימת שכנים של v ~~[illegible]~~ אחרי

20/20