



שפת C

הנדסאים וטכנאים מוסמכים – הנדסת תוכנה הנחיות לנבחן

א. משך הבחינה:

ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: חלק א': 15 נקודות

שאלה 1 – חובה.

חלק ב': 40 נקודות

שאלות 2-4: יש לענות על שתי שאלות בלבד.

ערך כל שאלה – 20 נקודות.

חלק ג': 45 נקודות

שאלות 5-8: יש לענות על שלוש שאלות בלבד.

ערך כל שאלה – 15 נקודות.

בסך הכול: 100 נקודות.

ג. חומר עזר מותר לשימוש:

1. מחשבון – אין להשתמש במחשב כף יד או במחשבון עם תוכנות או במחשבון המאפשר תקשורת חיצונית.

2. קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר. אין לצרף ספרים/חוברות עם פתרונות.

1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה ולוודא שהן מובנות.

2. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות.

3. יש לכתוב את התשובות במחברת הבחינה בעט בלבד, בכתב יד ברור.

4. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה.

5. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר על הכתוב קו כדי שלא ייבדק.

6. יש להסביר בפירוט כל תכנית שנכתבה. תכנית ללא הסבר מפורט לא תזכה לניקוד.

7. אם לדעתך חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לך להמשיך בפתרון השאלה. נמק את בחירתך.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!
ההנחיות בשאלון זה מנוסחות בלשון זכר, אך מכוונות לנבחנות ולנבחנים כאחד.

חלק א' (15 נקודות)

ענה על שאלה 1 – שאלת חובה (15 נקודות).

שאלה 1

נתונה התכנית הבאה בשפת C:

```
#include <stdio.h>
#define swap2(a,b)  a=a+b;b=a-b;a=a-b;

void swap1(int a,int b)
{
    int temp;
    temp=a;
    a = b;
    b = temp;
}

int main()
{
    unsigned char a,b,c;
    int x=5,y=10;

    printf("Section A:\n");
    swap1(x,y);
    printf("%d %d\n",x,y);
    swap2(x,y);
    printf("%d %d\n",x,y);

    printf("Section B:\n");
    b=5;
    a=++b;
    a += ++b;
    c= ~a & 0xf;
    printf("%d %d %d\n",a,b,c);
    return 0;
}
```

מה תדפיס התכנית? (Section A – 7 נקודות, Section B – 8 נקודות)
נמק את תשובתך באמצעות טבלת מעקב או הסבר החישובים.

חלק ב' (40 נקודות)

ענה על שתיים מבין השאלות 2-4 (לכל שאלה – 20 נקודות).

שאלה 2

כתוב פונקציה בשפת C, אשר מקבלת כפרמטר מחרוזת str. הפונקציה תחזיר מספר שלם.

הגדרת הפונקציה תהיה:

```
int func(char *str)
```

הפונקציה מקבלת מחרוזת str אשר מכילה תווים אלו בלבד:

- ספרות 0-9.
- האות האנגלית x
- רווחים

המחרוזת הנ"ל מייצגת מספרים באופן הבא:

- כל מספר מיוצג על-ידי תווים שבהם ספרות ברצף.
- מספרים שונים מופרדים על-ידי רווח אחד בלבד (אין צורך לבדוק רצף של רווחים).
- ישנה דרך קיצור לרישום מספרים והיא כוללת שימוש בתו x, שמייצג חזרה על אותה ספרה מספר פעמים, באופן הבא: DxN זו הספרה D, רשומה N פעמים (כאשר N הוא מספר בין 1 ל-9 כולל).
- למשל: אם רוצים לכתוב את המספר 2222, כלומר ארבע פעמים את הספרה 2, אפשר לכתוב $2x4$.
- ואת המספר 4255557 אפשר לכתוב $425x7$, כלומר אחרי 42 מופיעה הספרה 5 ארבע פעמים ובסוף הספרה 7.

הפונקציה תבצע את הפעולות הבאות:

- (10 נק') א. תבדוק את תקינות הקלט. אם הקלט לא תקין, תחזיר -1 ולא תדפיס דבר על המסך.
- אם הוא תקין, תמשיך לסעיף הבא. בדיקת התקינות כוללת שימוש רק בתווים חוקיים (ספרות, רווחים והתו 'x'). יש לבדוק גם שימוש נכון באות x (x תמיד חייב להופיע בין שתי ספרות).

הערה: מותר להניח כי המצביע str תקין, המחרוזת איננה ריקה ואין לבדוק זאת, אין צורך להתייחס לרצף של רווחים. וגם מותר להניח שאין רווח לפני המספר הראשון ואחרי המספר האחרון (ראה דוגמאות בעמוד הבא).

- ב. (10 נק') תדפיס את המספרים שהמחרוזת מייצגת לפי סדר הופעתם. כל מספר בשורה נפרדת (אין צורך לשמור אותם במשתנים, מספיק להדפיס אותם. בכל רווח יש לעבור שורה בהדפסה). הפונקציה תחזיר שלם שמייצג את מספר המספרים שהודפסו על המסך. מותר השימוש בכל פונקציות עזר ובספריות סטנדרטיות.

דוגמאות:

דוגמה 1:

```
str = "123 456 789"
```

הפונקציה תדפיס:

123

456

789

ותחזיר את המספר 3, כי הודפסו שלושה מספרים.

דוגמה 2:

```
str = "234 13x56 2x6"
```

הפונקציה תדפיס:

234

1333336

222222

ותחזיר את המספר 3.

דוגמה 3:

```
str = "234 3x5 2x6 x2"
```

הפונקציה תחזיר 1- כי המחרוזת אינה תקינה (ה־x מגיע אחרי תו רווח).

שאלה 3

נתונה ההגדרה הבאה:

```
typedef struct node
{
    int num;
    struct node *next1;
    struct node *next2;
} Node;
```

שמשמשת להגדרת רשימה מקושרת.

כתוב פונקציה בשפת C שמקבלת כקלט מערך של עשרה מספרים שלמים, ומחזירה מצביע לרשימה מקושרת של רשומות מסוג Node.

הגדרת הפונקציה תיראה כך:

```
Node *func (int num[10])
```

הפונקציה תקצה זיכרון לכל ערך במערך הקלט מסוג רשומת Node ותכניס את המספר לשדה num. מצביעי next1 יצביעו לאיבר הבא לפי סדר הקלט המקורי.

הפונקציה תמצא את הערך המקסימלי שבקלט ותעדכן את מצביעי next2 להצביע לאיבר זה. אם יש יותר מערך מקסימלי אחד, תצביע למופע הראשון שלו.

מלבד מצביע next2 של האיבר המקסימלי, שבו יעודכן המצביע להצביע על האיבר עם הערך המינימלי (אם יש יותר מערך מינימלי אחד יצביע למופע הראשון שלו).

הפונקציה תחזיר מצביע לאיבר הראשון ברשימה.

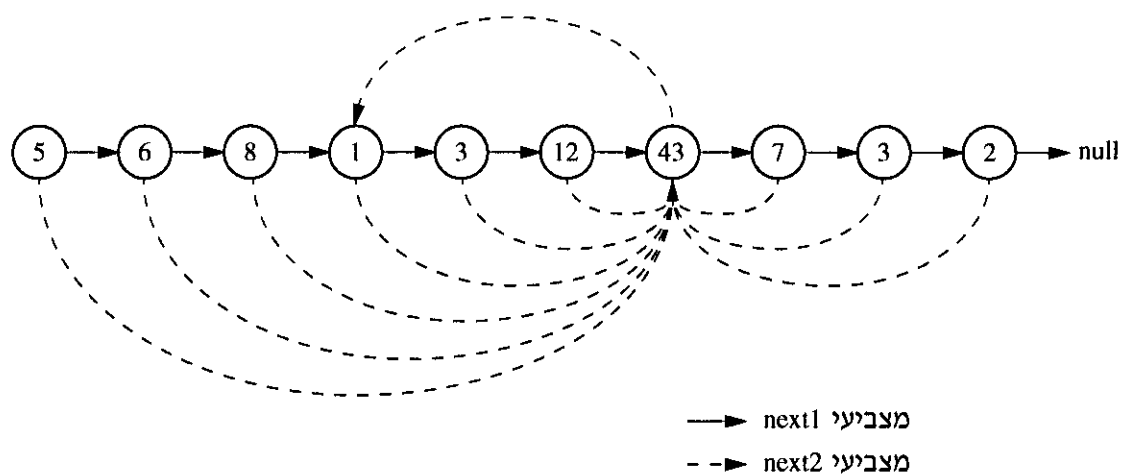
המשך השאלה בעמוד הבא:

דוגמה:

עבור:

num = {5,6,8,1,3,12,43,7,3,2}

נקבל את הרשימה הבאה:



איור לשאלה 3

שאלה 4

כתוב פונקציה המקבלת כפרמטר מערך דו-ממדי – מטריצה מסוג int בגודל $N \times N$ של מספרים 0-1 בלבד.

הפונקציה תבדוק אם המטריצה שהתקבלה היא מטריצה שאיבריה הם 0-1 בלבד. אם לא – תחזיר 0, אם כן – תמשיך.

הפונקציה תדפיס את האינדקסים של איברים שערכם הוא 0 והם "מוקפים" ב-1-ים (מוקף ב-"1" מעל האיבר, מתחתיו, מצדדיו ובאלכסונים).

עבור איבר שנמצא בקצה המטריצה ייבדקו רק איברים שקיימים צמודים אליו. למשל לאיבר בשורה 0 של המטריצה ייבדקו השכנים לידו באותה שורה ומתחתיו. ראה דוגמה.

אם אין איברים כאלו, הפונקציה תדפיס הודעה מתאימה.

הפונקציה תחזיר 1 (במצב שהקלט תקין).

דוגמה (עבור מטריצה 4×4 כלומר $N = 4$):

0	1	1	0
1	1	1	1
1	0	1	0
1	1	1	0

בדוגמה זו הפונקציה תדפיס את האינדקסים הבאים:

0,0

0,3

2,1

בשלושת המקומות הללו יש 0 שמוקף ב-1-ים בלבד.

המקומות 2,3 ו-3,3 לא מודפסים כי האפסים במקומות האלה לא מוקפים ב-1-ים בלבד.

הפונקציה תחזיר 1.

חלק ג' (45 נקודות)

ענה על שלוש מבין השאלות 5-8 (לכל שאלה - 15 נקודות).

שאלה 5

נתונה התכנית הבאה הכתובה בשפת C:

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

int func(int vec[],int c,int k)
{
    int i,j,temp,temp_i;
    for(i=0;i<k;i++)
    {
        temp_i = i;
        for (j=i+1;j<c;j++)
            if (vec[j]<vec[temp_i])
                temp_i=j;
        temp = vec[i];
        vec[i] = vec[temp_i];
        vec[temp_i] = temp;
    }
    return (vec[k-1]);
}

int main()
{
    int vec[] = {1,20,2,30,3,40,4,50,5,60,6,8,10,12,14,-1};
    int k,c,ans;

    printf("\n");
    for (c=0;vec[c]>0;c++) printf("%d ",vec[c]);
    printf("\n");
```



```
printf("enter an integer number from [1-%d]: ",c);
scanf ("%d",&k);
if (k<=c)
{
    ans = func(vec,c,k);
    printf("the number is: %d",ans);
}
else
    printf( "error");
return 0;
}
```

- א. (3 נק') מהו ערכו של c אחרי הרצת התכנית? מה מייצג משתנה זה?
- ב. (3 נק') עבור הכנסת הערך 5 מהמשתמש (למשתנה k) מה תדפיס התכנית?
- ג. (3 נק') עבור הכנסת הערך 8 מהמשתמש (למשתנה k) מה תדפיס התכנית?
- ד. (3 נק') עבור הכנסת הערך 12 מהמשתמש (למשתנה k) מה תדפיס התכנית?
- ה. (3 נק') הסבר מה מבצעת הפונקציה func.

שאלה 6

נתונה התכנית הבאה הכתובה בשפת C:

```
#include <stdio.h>

int func(int a,int b,int index)
{
    if (index <= 2) return 1;
    if (index == 3) return a+b;
    else return (func(b,a+b,index-1));
}

int main()
{
    int i;

    printf("\nThe numbers are: \n");
    for (i=1;i<=10;i++)
        printf("%d ",func(1,1,i));

    return 0;
}
```

- (8 נק') א. עבור הרצת $\text{func}(1,1,5)$, תאר כמה פעמים נקראת הפונקציה עד שתחזיר את הערך.
רשום את ערכי הפרמטרים של ההרצות של func ואת הערך שהפונקציה func תחזיר לבסוף.
- (7 נק') ב. מה תדפיס התכנית?

שאלה 7

נתונה התכנית הבאה הכתובה בשפת C:

```
#include <stdio.h>

#define N 8

int main()
{
    unsigned long int num1,num2;
    unsigned char ch1[N],ch2[N];
    int i;

    num1 = 0x1d3ba4c2;
    printf("\noriginal number is: %lx\n",num1);
    for (i=0;i<N;i++)
    {
        ch1[i]=num1&0xf;
        num1 = num1>>4;
    }
    for (i=0;i<N;i+=2)
    {
        ch2[i] = ch1[N-1-i];
        ch2[i+1] = ch1[i];
    }
    num2=0;
    for (i=0;i<N;i++)
    {
        num2 = num2 << 4;
        num2 += ch2[i];
    }
    printf( "new number is: %lx\n",num2);
    return 0;
}
```

יש להציג בתשובה את כל הערכים בבסיס hex.

5 נק' א. מהי תכולת המערך ch1 בסיום הריצה?

5 נק' ב. מהי תכולת המערך ch2 בסיום הריצה?

5 נק' ג. מה תדפיס התכנית בסיום הריצה?

שאלה 8

נתונה התכנית הבאה הכתובה בשפת C:

```
#include <stdio.h>

#define N 5
#define M 3

int main()
{
    int mat1[N][N] = {{ 1,2,3,4,5},
                      { 1,2,3,4,5},
                      { 1,2,3,4,5},
                      { 1,2,3,4,5},
                      { 5,5,5,5,5}};

    int mat2[N-M+1][N-M+1] = {0};
    int i,j,t,k,s;

    s = N-M+1;

    printf("\n");
    for (i=0;i<s;i++)
    {
        for (j=0;j<s;j++)
        {
            for (t=0;t<M;t++)
            for (k=0;k<M;k++)
                mat2[i][j]+=mat1[i+t][j+k];
            printf("%d ",mat2[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

(10 נק') א. מה תדפיס התכנית?

(5 נק') ב. תאר מה תבצע התכנית כאשר נשנה את M ל-4. אין צורך לחשב את ההדפסה.
סרטט איור של `mat1` וסמן בו את איברי `mat1` המשמשים לחישוב כל אחד מאיברי `mat2`.

בהצלחה!

© כל הזכויות שמורות למה"ט