



## 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.
- ד. הוראות כלליות:
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 להציג פתרון מלא ומנומק, כולל חישובים במקרים שצריך. תשובה סופית ללא דרך הפתרון לא תזכה לניקוד.
  7. יש להסביר בפירוט כל תכנית שנכתבה. תכנית ללא הסבר מפורט לא תזכה לניקוד.
  8. 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.

בשאלון זה 11 עמודים.

**בהצלחה!**

## חלק א' (15 נקודות)

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

### שאלה 1

נתונה התכנית הבאה בשפת C:

```
#include <stdio.h>

int main()
{
    int x=3;
    int a;

    printf("\n");
    for (a=0 ; a<3 ; a++)
    {
        if (!a || x++)
            x--;
        printf("%d ",x);
    }

    printf("\n");
    x=3;
    for (a=1 ; a<3 ; a++)
    {
        if (!a || x++)
            x--;
        printf("%d ",x);
    }

    printf("\n");
    x=13;
    for(a=7 ; a<x ; a-=2)
    {
        x = x % a;
        printf("%d ",x);
    }

    return 0;
}
```

מה תדפיס התכנית? (כל הדפסה 5 נק')

נמק את תשובתך באמצעות טבלת מעקב לביצוע הלולאות.

## 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 מצביע למחרוזת חדשה. הגדרת הפונקציה תהיה:

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

הפונקציה מקבלת מחרוזת `str` אשר מכילה תווים אלו בלבד:

- אותיות קטנות באנגלית
- אותיות גדולות באנגלית
- רווחים

הפונקציה תבצע את הפעולות הבאות:

**(5 נק')** א. תבדוק את תקינות הקלט. אם הקלט לא תקין הפונקציה תחזיר `NULL`.

אם הוא תקין הפונקציה תמשיך לסעיף הבא.

**הערה:** מותר להניח כי המצביע `str` תקין ואין לבדוק זאת.

**(15 נק')** ב. הפונקציה תעביר את כל האותיות הגדולות שבמחרוזת לסוף המחרוזת, בסדר המקורי של האותיות, ותחליף את הרווחים בתו `'#'`. יש להקצות זיכרון למחרוזת החדשה ולהחזיר אליה מצביע (לא נדרש לבדוק את ההקצאה הדינמית, הנח כי הצליחה).

הפונקציה לא תשנה את הערכים במחרוזת `str`.

דוגמה:

```
str = "helloH wELlorld 0"
```

אחרי ביצוע הפונקציה המחרוזת החדשה תיראה כך:

```
str_new = "hello#world#HELLO"
```

אין מגבלה על מספר הפעמים שאפשר לסרוק את המחרוזת המקורית.

**הערה:** אפשר להשתמש בפונקציות מספריית `string.h` וגם מספריית `ctype.h`.

## שאלה 3

נתונה ההגדרה הבאה:

```
typedef struct node
{
    int num;
    struct node *next;
} Node;
```

ההגדרה משמשת להגדרת רשימה מקושרת. בכל איבר יש מספר שלם בשם num ומצביע לאיבר הבא בשם next.

כתוב פונקציה בשפת C שקולטת מצביע לרשימה מקושרת של איברים מסוג Node ופרמטר נוסף שלם בשם number, ושתחזיר מצביע לרשימה. הגדרת הפונקציה תיראה כך:

```
Node *func(Node *head, int number)
```

הפונקציה תבצע את הפעולות הבאות:

- תבדוק את מספר האיברים שיש ברשימה שהתקבלה, ותדפיס מספר זה.
- אם מספר זה גדול מהפרמטר number, התכנית תשאיר ברשימה המקורית את number האיברים האחרונים של הרשימה (כלומר תמחק את האיברים הראשונים).
- הערה: מותר להניח כי  $number > 0$  ולא לבדוק זאת בתוך הפונקציה.
- אם מספר זה קטן מהפרמטר number, התכנית תוסיף איברים בתחילת הרשימה עם ערך  $num = 0$  כדי שתהיה רשימה באורך number.
- הפונקציה תחזיר מצביע לראש הרשימה החדשה.
- לא נדרש לבדוק את ההקצאה הדינאמית, הנח כי הצליחה.

המשך השאלה בעמוד הבא:

דוגמאות:

**עבור רשימה מקורית:**

**Head → 4 → 9 → 5 → 10 → 5 → 3 → NULL**

עבור: func(head,4) שרץ על הרשימה המקורית

התכנית תדפיס:

Number of elements: 6

ותשנה את הרשימה המקורית ותחזיר אליה מצביע שייראה כך:

Head → 5 → 10 → 5 → 3 → NULL

כלומר נמחקו שני האיברים הראשונים.

עבור: func(head,8) שרץ על הרשימה המקורית

התכנית תדפיס:

Number of elements: 6

ותשנה את הרשימה המקורית ותחזיר אליה מצביע שייראה כך:

Head → 0 → 0 → 4 → 9 → 5 → 10 → 5 → 3 → NULL

הוספו שני איברים לתחילת הרשימה עם הערך num = 0.

## שאלה 4

כתוב פונקציה המקבלת כפרמטר מערך דו־ממדי - מטריצה מסוג  $n \times n$  בגודל  $n$  של ספרות 0-9 בלבד.

הפונקציה תבדוק אם המטריצה שהתקבלה היא מטריצה המקיימת את התנאים הבאים:

- כל הספרות הן 0-9 בלבד.
  - בכל שורה בנפרד יש מספרים זוגיים או אי־זוגיים בלבד. כלומר אין בשורה ערבוב של מספרים זוגיים ואי־זוגיים יחד.
  - סכום הספרות של כל עמודה קטן מ־ $5 \times N$ .
- הפונקציה תחזיר 1 אם המטריצה מקיימת את התנאים, ו־0 אם לא.
- דוגמה (על מטריצה  $4 \times 4$  כלומר  $N = 4$ ):

0	2	8	4
1	3	1	7
6	2	2	2
3	3	3	3

במטריצה זו כל הספרות הן 0-9, ומתקיים התנאי הראשון.

המספרים בשורה 0 הם כולם זוגיים, בשורה 1 כולם אי־זוגיים, בשורה 2 זוגיים, בשורה 3 אי־זוגיים, כלומר מתקיים התנאי השני.

סכומי העמודות:

$$\text{עמודה 0: } 0 + 1 + 6 + 3 = 10$$

$$\text{עמודה 1: } 2 + 3 + 2 + 3 = 10$$

$$\text{עמודה 2: } 8 + 1 + 2 + 3 = 14$$

$$\text{עמודה 3: } 4 + 7 + 2 + 3 = 16$$

כל הסכומים קטנים מ־ $20$  ( $4 \times 5$ ) לכן התנאי השלישי גם מתקיים ולכן הפונקציה תחזיר 1.

אם נשנה את השורה האחרונה כך שהספרה 7 תופיע בכל האיברים במקום 3 כפי שיש בדוגמה למעלה, התנאים 1 ו־2 יתקיימו אולם סכום האיברים בעמודה 3 יהיה  $4 + 7 + 2 + 7 = 20$ , כלומר התנאי השלישי לא יתקיים והפונקציה תחזיר 0.

## 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>

#define N 6

int func (int vec[],int num)
{
    int a;
    if (num>N-1)
        return 0;
    a=vec[num];
    if ( (num!=(N-1)))
        if ((vec[num]>vec[num+1]))
            num = N;
    return (a+func(vec,num+1));
}

int main()
{
    int vec[N];
    int res,i;
    for (i=0;i<N;i++)
        vec[i] = i+1;
    res = func(vec,0);
    printf("\n%d\n",res);
    for (i=3;i<N;i++)
        vec[i] = (3*i)%N;
    res = func(vec,0);
    printf("\n%d\n",res);
    return 0;
}
```

(5 נק') א. מה מבצעת הפונקציה func?

(10 נק') ב. מה תדפיס התכנית?

כתוב במחברת את תכולת הווקטור vec לפני כל הרצת func (בכל פעם אחרי לולאת ה-for).

## שאלה 6

נתונה התכנית הבאה הכתובה בשפת C:

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

char *func(char *str,int num1,int num2)
{
    int szel;
    char *res,tmp;

    szel = strlen(str);
    res = (char *) malloc(sizeof(char)*(szel-num2+1));
    if (res == NULL)
        return NULL;
    if ((num1+num2)>szel)
        return NULL;
    tmp=str[num1];
    str[num1] = '\0';
    strcpy(res,str);
    str[num1] = tmp;
    strcat(res,str+num2+num1);
    return res;
}

int main()
{
    char str[] = "I_feel_good_bad";
    char *res;

    res = func(str,7,5);
    printf("\n%s\n",res);
    free(res);

    res = func(str,11,4);
    printf("\n%s\n",res);
    free(res);

    return 0;
}
```

(5 נק') א. מה מבצעת הפונקציה func?

(10 נק') ב. מה תדפיס התכנית? נמק את תשובתך.



## שאלה 7

נתונה התכנית הבאה הכתובה בשפת C:

```
#include <stdio.h>

char func( unsigned char num1)
{
    unsigned char temp1,temp2;
    temp1 = temp2 = num1;

    temp1 = temp1 & 0x88;
    temp1 = temp1 >> 3;

    temp2 = temp2 & 0x11;
    temp2 = temp2*8;

    temp2 = temp2|temp1;
    num1 = num1&0x66|temp2;

    return num1;
}

int main()
{
    unsigned char num;

    num = 0x57;
    printf("\n%x ->  ",num&0xff);
    num = func(num);
    printf("%x \n",num&0xff);

    num = 0xab;
    printf("\n%x ->  ",num&0xff);
    num = func(num);
    printf("%x\n",num&0xff);

    return 0;
}
```

(5 נק') א. מה מבצעת הפונקציה func?

(10 נק') ב. מה תדפיס התכנית? נמק את תשובתך.

## שאלה 8

נתונה הפונקציה הבאה הכתובה בשפת C:

```
#include <stdio.h>

#define N 3

int func (int mat[2*N][2*N])
{
    int i,j;
    int index,sum[N]= {0};
    for (i=0;i<N ;i++)
    {
        for (j=0; j < ((N-i)*2-1);j++)
        {
            index = 2*N-i-1;
            sum[i] += mat[i][i+j]
                    + mat[i+j][index]
                    + mat[index][index-j]
                    + mat[index-j][i];
        }
        if (i!=0)
            if (sum[i]!=sum[i-1])
                break;
    }
    return ( i != N);
}

int main()
{
    int mat1[2*N][2*N] = {{ 1,0,1,1,0,1},
                           { 8,5,3,3,6,1},
                           { 0,3,9,9,1,1},
                           { 1,3,8,8,1,0},
                           { 1,2,3,1,3,6},
                           { 2,2,2,2,2,2}};
```

המשך התכנית בעמוד הבא:

```
if (func(mat1))
    printf("\nNo!!!");
else printf("\nYes!!!\n");
return 0;
}
```

(10 נק') א. מה מבצעת הפונקציה func? מה יהיו ערכי המערך sum בסיום הריצה?

(5 נק') ב. מה תדפיס התכנית?

**בהצלחה!**

© כל הזכויות שמורות למה"ט