



## 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.

ד. הוראות כלליות:

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.

בהצלחה!

## חלק א' (15 נקודות)

ענה על שאלה 1 - שאלת חובה

שאלה 1 (15 נקודות)

נתונה תוכנית הבאה בשפת C :

```
#include <stdio.h>

#define SUM_OF_3(x,y,z) x+y+z

#define AVRГ_OF_3(x,y,z) (x+y+z)/3

#define AVRГ1_OF_3(x,y,z) SUM_OF_3(x,y,z)/3

void main()
{
    int a=10, b=8, c=7;

    printf("The average of %d,%d,%d is %d\n",a,b,c,AVRГ_OF_3(a,b,c));

    a<<=1;
    b>>=1;
    c^=1;
    printf("The average of %d,%d,%d is %d\n",a,b,c,AVRГ_OF_3(a,b,c));

    a>>=1;
    b<<=1;
    c^=1;
    printf("The average of %d,%d,%d is %d\n",a,b,c,AVRГ1_OF_3(a,b,c));
}
```

בתוכנית יש שלוש פקודות הדפסה. מה תדפיס התוכנית עבור כל אחת מפקודות ההדפסה הללו? כתוב טבלת מעקב ופרט את החישובים.

## חלק ב' (40 נקודות)

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

### שאלה 2 (20 נקודות)

לפעמים בהקלדה אנו מעוניינים להחליף מילה או צירוף מילים נפוץ במחרוזת קצרה יותר (קוד) ובכך לצור קיצור דרך. למשל במקום "mobile phone" אנו יכולים לצור קוד "123" (לא חייב להיות מורכב מספרות) שהוא גם קצר וגם נוח להקלדה. נניח שבחרנו קוד מסוים עבור צירוף מילים מסוים והקלדנו את הטקסט. בהדפסתו אנו רוצים שהקוד שבחרנו יוחלף בחזרה בצירוף המילים המקורי.

כתוב פונקציה:

```
void PrintDecoded(char* text, char* str2insert)
```

אשר מקבלת שני מצביעים לתחילת שתי מחרוזות כאשר המחרוזות הראשונה מורכבת מקידומת (ראה הסבר בהמשך) וטקסט ובו מופעי קוד אשר מחליפים את צירוף המילים המקורי. המחרוזת השנייה מהווה צירוף מילים מקורי. כלומר אם נתבונן בדוגמה הנ"ל, בתוך מחרוזת text מייד אחרי הקידומת המוסברת בהמשך כל מופע של "123" מייצג מחרוזת "mobile phone". והמחרוזת המקורית הזו נמצאת בפרמטר השני: str2insert.

הקידומת במחרוזת הראשונה מכילה את הקוד המייצג את המחרוזת השנייה מוקף בשני תווים זהים אך לא ידוע מראש מה הם. למשל הקידומת של text אשר מכיל קוד "123" יכולה להראות כך:

#123#.....

או למשל כך:

@123@.....

עליך לממש את הפונקציה PrintDecoded כך שתודפס המחרוזת הראשונה ללא קידומת המכילה את הקוד הנ"ל ובה כל מופע של הקוד יוחלף במחרוזת השנייה.

למשל אם הקוד "123" מחליף את צירוף המילים "mobile phone" והמחרוזת הראשונה text נראית כך:

"#123#Choose carefully your 123 because in these days it is more than just 123."

הפונקציה תדפיס:

Choose carefully your mobile phone because in these days it is more than just mobile phone

הערה 1: אפשר להניח כי אורך הקוד לא עולה על 10 תווים אך אי אפשר להניח שום הנחות לגבי אורך המחרוזת text ו-str2insert.

הערה 2: מותר ואף מומלץ להשתמש בפונקציות המוגדרות בספרייה string.h

**שאלה 3 (20 נקודות)**

נתונה ההגדרה הבאה :

```
typedef struct node
{
    int key;
    struct node *next;
} NODE;
```

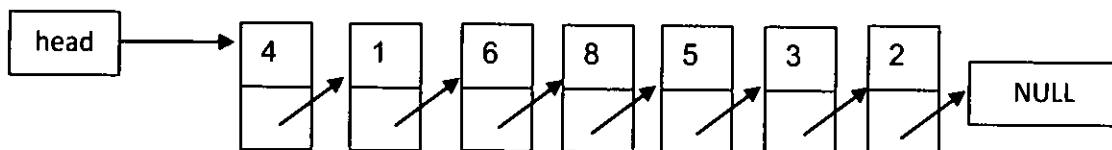
שמשמשת להגדרת רשימה מקושרת. כתוב פונקציה :

**NODE\* Reorder(NODE\* head)**

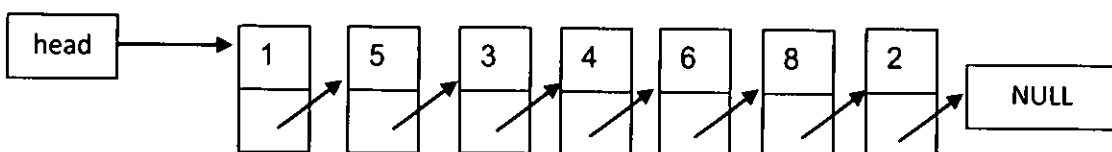
אשר קולטת מצביע לראש הרשימה המקושרת ובונה רשימה חדשה מתוך איברי הרשימה המקורית בדרך שמתוארת בהמשך. הפונקציה תחזיר מצביע לראש הרשימה החדשה. בניית הרשימה החדשה מבוססת על כללים הבאים :

- לכל איבר ברשימה המקורית מותאם איבר ברשימה החדשה ;
- ברשימה החדשה האיברים בעלי מפתח אי-זוגי יופיעו לפני כל האיברים בעלי מפתח זוגי ;
- ברשימה החדשה סדר בין האיברים בעלי מפתח אי-זוגי יישאר כמו שהיה ברשימה המקורית ;
- ברשימה החדשה סדר בין האיברים בעלי מפתח זוגי יישאר כמו שהיה ברשימה המקורית.

למשל אם הרשימה המקורית הייתה כזו :



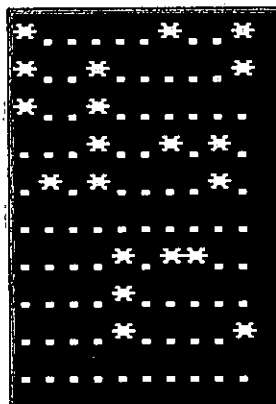
הרשימה העדשה תראה כך :



**שאלה 4 (20 נקודות)**

בשלב היערכות במשחק הקלאסי "קרב הצוללות" יש למקם את 10 הצוללות באורך 1 עד 4 במטריצה בגודל 10x10 במאונך או במאונך כך שיהיה רווח של משבצת אחת לפחות בין כל הצוללות מכל כיוון, כולל באלכסון. על מנת לאפשר למחשב למקם את הצוללות באופן אוטומטי, צריך בין השאר לוודא שהמשבצת והכיוון שנבחרו באקראי למיקום הצוללת נתון אכן משאירים רווח של משבצת אחת לפחות בין הצוללת הנוכחית לשאר הצוללות ושהצוללת הנוכחית לא "גולשת" מגבולות המטריצה.

להלן דוגמה של מיקום חוקי של 10 הצוללות:



נניח שהוגדרו הגדרות הבאות:

```
#define MSIZE 10
#define VERTICAL 0
#define HORIZONTAL 1
#define SHIP '*'
#define EMPTY '.'
```

לוח המשחק יוצג על ידי מטריצה בגודל MSIZExMSIZE מסוג char. צוללת תסומן במטריצה על ידי תו SHIP, משבצות פנויות יסומנו על ידי תו EMPTY וכיוון יקבל ערך HORIZONTAL או VERTICAL.

כתוב פונקציה:

```
int IsAvailable(char board[MSIZE][MSIZE], int ship_size, int row, int col, int dir)
```

אשר מקבלת לוח – מטריצה מסוג char בגודל MSIZExMSIZE המכילה צוללות שכבר מוקמו, אורך הצוללת הנוכחית, קואורדינטה של משבצת שהוגרלה באקראי שתהווה קצה שמאל עליון של הצוללת וכיוון שיכול להיות HORIZONTAL או VERTICAL בהתאם להגדרות הנ"ל. הפונקציה תבדוק האם ניתן למקם את הצוללת לפי כללי המשחק ותחזיר 1 אם אכן ניתן למקם אותה. אחרת הפונקציה תחזיר 0.

## חלק ג' (45 נקודות)

ענה על שלוש מבין השאלות 5-9 (לכל שאלה – 15 נקודות).

שאלה 5 (15 נקודות)

נתונה תכנית הבאה :

```
void func(int* a, int s, int num)
{
    int* p = a + num%2;
    int z = (s+1-num%2)/2;
    int i,j,m,x;

    for(i=0;i<z;i++)
    {
        for(j=2,m=0;j<(z-i)*2;j+=2)
            if(*(p+j)>*(p+m))
                m = j;
        x = *(p+(z-i-1)*2);
        *(p+(z-i-1)*2) = *(p+m);
        *(p+m) = x;
    }
}

void main()
{
    int data[] = {5,2,7,6,1,-3,9,0,4,-2,8,3};
    int i, num;

    printf("Enter a number: ");
    scanf("%d",&num);

    func(data,sizeof(data)/sizeof(data[0]),num);

    printf("Data is:\n");
    for(i=0;i<sizeof(data)/sizeof(data[0]);i++)
        printf("%d ", *(data+i));
}
```

(א) 5 נקודות) מה מבצעת פונקציה func?

(ב) 5 נקודות) מה תדפיס התוכנית על מספר קלט 5?

(ג) 5 נקודות) מה תדפיס התוכנית על מספר קלט 6?

שאלה 6 (15 נקודות)

נתונה תכנית הבאה:

```

void func(char* s)
{
    char tmp;
    int z;

    if(*s!=' ' && *s!='\0')
    {
        for(z=0;s[z]!=' ' && s[z]!='\0'; z++);
        z--;
        tmp = *s;
        *s = *(s+z);
        *(s+z) = ' ';
        func(s+1);
        *(s+z) = tmp;
    }
}

void main()
{
    char str[] = "good luck on exam";
    char* p;

    p=str;

    while(*p!='\0')
    {
        while(*p==' ') p++;
        func(p);
        while(*p!=' ' && *p!='\0') p++;
    }
    printf("%s\n",str);
}

```

א) (5 נקודות) מה מבצעת פונקציה func?

ב) (10 נקודות) מה תדפיס התוכנית?

שאלה 7 (15 נקודות)

נתונה תכנית הבאה:

```

#define MS 5

void func(int m[MS][MS])
{
    int a,b,c,d,p,q;

    for(p=0;p<MS/2;p++)
        for(q = p;q<MS-p-1;q++)
        {
            a = m[p][q];
            b = m[q][MS-p-1];
            c = m[MS-p-1][MS-q-1];
            d = m[MS-q-1][p];
            m[p][q] = d;
            m[q][MS-p-1] = a;
            m[MS-p-1][MS-q-1] = b;
            m[MS-q-1][p] = c;
        }
}

void main()
{
    int m[MS][MS];
    int i,j;

    for(i=0;i<MS;i++)
        for(j=0;j<MS;j++)
            m[i][j] = i;

    func(m);

    for(i=0;i<MS;i++)
    {
        for(j=0;j<MS;j++)
            printf("%d ",m[i][j]);
        printf("\n");
    }
}

```

- א) (5 נקודות) מה מבצעת פונקציה `func`?
- ב) (10 נקודות) מה תדפיס התוכנית?



שאלה 8 (15 נקודות)

נתונה תכנית הבאה :

```
void Func1(int *px, int* py,unsigned char c)
{
    if(*py<sizeof(long int)/sizeof(unsigned char))
    {
        (*px)<<=8;
        (*px) |= c;
        (*py)++;
    }
}

char Func2(int *px, int* py)
{
    unsigned char tmp='\0';
    if(*py>0)
    {
        tmp = (*px)>>8;
        (*px)>>=8;
        (*py)--;
    }
    return tmp;
}

void main()
{
    unsigned char ch;
    int x;
    int y=0,opr;

    printf("Enter an operation (0/1) or -1 to stop: ");
    scanf("%d",&opr);
```

```

while(opr!=-1)
{
    if(opr==0)
    {
        printf("Enter a char: ");
        fflush();
        scanf("%c",&ch);
        Func1(&x,&y,ch);
    }
    else if(opr==1)
        ch = Func2(&x,&y);
    printf("Enter an operation (0/1) or -1 to stop: ");
    scanf("%d",&opr);
}

while(y>0)
{
    y--;
    ch = x>>(8*y);
    printf("%c",ch);
}
printf("\n");
}

```

א) (7 נקודות) מה תדפיס התוכנית על קלט הבא (משמאל לימין):

0 a 0 b 0 c 0 d -1

ב) (8 נקודות) מה תדפיס התוכנית על קלט הבא (משמאל לימין):

0 a 0 b 1 0 c -1

- ב ה צ ל ח ה -

© כל הזכויות שמורות למה"ט