



1



מערכות תוכנה להנדסאי הנדסת אלקטרוניקה

הנחיות לנבחן

- א. משך הבחינה: ארבע שעות.
- ב. מבנה השאלון ומפתח הערכה: בשאלון 10 שאלות, יש לענות על 5 שאלות. משקל כל שאלה 20 נקודות.
- ג. חומר עזר מותר בשימוש: מחשבון, למעט מחשב נישא (מחשב מחברת או דומה), קלסר אחד עם חומר ההרצאות, שני ספרים. אין (אסור) להעביר חומר עזר, ספרים ומחשבוניס בין הנבחנים.
- ד. הוראות מיוחדות:
 1. יש להשאיר את הדף הראשון במחברת הבחינה ריק. בגמר הבחינה יש לרשום בעמוד זה את מספרי השאלות אותן ברצונך שמעריך הבחינה יבדוק. לא יבדקו שאלות עודפות על הנדרש.
 2. יש להתחיל כל תשובה בראש עמוד חדש. יש להתחיל את הפתרון במשפט "פתרון שאלה א'", x מספר השאלה שבשאלון. אין למספר את עמודי מחברת הבחינה.
 3. יש להעביר קו אלכסוני על הדפים או חלקי השאלה אותם אין ברצונך שהמעריך יקרא.
 4. אם לדעתך חסר נתון או קיים נתון שגוי, עליך לציין זאת במפורש ולהניח הנחה סבירה ומנומקת שתאפשר את המשך הפתרון.
 5. אם צוין במחברת הבחינה שאלות המכילות השלמת טבלה ניתן לפתור בשאלון הבחינה ולצרף את הדף למחברת הבחינה. (ציין מספר מכללה, מספר נבחן ומס' ת.ז.).

בהצלחה !

חלק א': תוכנה C++

שאלה מספר 1 (20 נקודות)

צופן קיסר הוא צופן להצפנת משפטים. בוחרים מספר t ($0 < t < 26$), ומחליפים כל אות באות הגדולה ממנה בערך t , באופן מחזורי. זכור כי באנגלית יש 26 אותיות.

(לדוגמה: עבור $t=3$, $A \rightarrow D, B \rightarrow E, \dots, X \rightarrow A, Y \rightarrow B, Z \rightarrow C$)

א) (10 נקודות) יש לכתוב מחלקה המחזיקה מחרוזת תווים באורך מקסימלי 80. כמו כן, יש להגדיר את פונקציות המחלקה הבאות:

- בנאי ברירת מחדל (default constructor) – שאינו מקבל פרמטרים
- בנאי (constructor) המקבל פרמטר מסוג `char *`
- פונקציה המאפשרת לקלוט קלט אל המחרוזת
- פונקציה המאפשרת להדפיס את המחרוזת לפלט
- פונקציה `encrypt` המקבלת כפרמטר את t ומצפינה את המחרוזת לפי צופן קיסר.

שים לב: בסעיף זה אין לממש פונקציות, אלא רק לכתוב prototypes שלהן.

ב) (10 נקודות) יש לממש את הפונקציה `encrypt` המצפינה את המחרוזת באמצעות צופן קיסר. ההצפנה תעשה על ידי הזזה של כל אות אנגלית גדולה במחרוזת, בערך t . (לדוגמה: אם $t=3$, המחרוזת "ABXYa" תוצפן ל "DEABa").

שאלה מספר 2 (20 נקודות)

לפניך מספר מקטעי תוכניות בשפת ++C. i, j משתנים מסוג `int`. `str` משתנה מטיפוס `char*` (באורך מספיק). בכל אחד מהסעיפים יש לרשום באופן ברור את הפלט בסיום הקטע. כמו כן יש לנמק כיצד הגעת/ה לתשובה זו.
(א) (7 נקודות) מה יודפס בסיום הקטע הבא?

```
strcpy(str, "ABCD");
i=0;
str[++i]='\0';
cout<<str<<endl;
```

(ב) (6 נקודות) מה יודפס בסיום הקטע הבא?

```
for(i=2; i<5; ++i)
    for(j=i; j<2*i; ++j)
        if( !(i%j) )
            cout<<i<<' ';
cout<<endl;
```

(ג) (7 נקודות) מה יודפס בסיום הקטע הבא?

```
strcpy(str, "ABCD");
i=str[1]-'A';
cout<<(str+i)<<endl;
```

שאלה מספר 3 (20 נקודות)
נתונה מחלקה

```
class Image{
    bool M[N][N];
public:
    void set();
    void print() const;
    int similar(const Image &im) const;
};
```

המחלקה מייצגת מטריצה של ערכים בוליאניים (כלומר, מטריצה המכילה רק את הערכים 0 או 1). N הוא קבוע המוגדר בתוכנית.
נגדיר "מקדם דמיון" בין שתי מטריצות כמספר האיברים הזהים בשתי המטריצות. בדוגמא הבאה מצויירות שתי מטריצות בגודל 3X3 (כלומר N=3). כפי שניתן לראות מהציור, יש 4 איברים זהים (אלו המסומנים בעיגול) ולכן מקדם הדמיון הוא 4.

$$\begin{pmatrix} 1 & \textcircled{1} & 1 \\ \textcircled{1} & 0 & \textcircled{0} \\ 1 & 0 & \textcircled{0} \end{pmatrix}, \begin{pmatrix} 0 & \textcircled{1} & 0 \\ \textcircled{1} & 1 & \textcircled{0} \\ 0 & 1 & \textcircled{0} \end{pmatrix}$$

א) (12 נקודות) יש לממש את הפונקציה similar. פונקציה זו מחזירה את מקדם הדמיון בין האובייקט המפעיל לבין הפרמטר שלה (im).

ב) (8 נקודות) יש לכתוב את הפונקציה:

```
int findSimilar(const Image &im, const Image imarr[], int size)
```

הפונקציה מקבלת שלשה פרמטרים: אובייקט (im), מערך אובייקטים (imarr) וגודל מערך האובייקטים (size). הפונקציה תחזיר את האינדקס של האובייקט במערך imarr הדומה ביותר לאובייקט im (כלומר, שמקדם הדמיון ביניהם מקסימאלי).

שאלה מספר 4 (20 נקודות)

כתוב לכל אחד מהסעיפים הבאים פונקציה (נפרדת!) ב ++c הבודקת / מדפיסה את המבוקש. כל אחת מהפונקציות מקבלת כפרמטר מערך שלמים (int) ואת גודלו. שים לב! בסעיף ב' יש לכתוב פונקציה רקורסיבית.

א) (6 נקודות) הפונקציה תחזיר true אם יש במערך (המועבר כפרמטר) שני מספרים סמוכים (כלומר בעלי אינדקסים עוקבים) שסכומם 17, אחרת תחזיר false.

ב) (6 נקודות) את סכום האברים הזוגיים במערך (סכום כל האברים שערכם זוגי). סעיף זה (ב') - חובה לממש ברקורסיה.

ג) (8 נקודות) לכל איבר במערך יש שני שכנים, אחד מימין ואחד משמאל (מלבד לאבר הראשון והאחרון שלהם יש רק שכן אחד). יש להדפיס את כל האברים הגדולים משני שכניהם (האבר הראשון והאחרון לא יודפסו לעולם).

שאלה מספר 5 (20 נקודות)

נתונה התוכנית

```

#include <iostream.h>
class vector2{
    double x, y;
public:
    vector2(double a, double b){ x=a; y=b;}
    virtual void print(){ cout<<x<<" "<<y<<endl; }
    void scale(double t){ x*=t; y*=t; }
    double getx(){return x;}
    double gety(){return y;}
};
class complex: public vector2{
public:
    complex(double a, double b):vector2(a,b){}
    virtual void print(){ cout<<getx()<<" + i*"<<gety()<<endl; }
    void scale(double t){ vector2::scale(t); print(); }
};
void main(){
    vector2 v1(1.0,0.5), *v2;
    v2=new complex(5.5,3.0);
    v1.print();
    v2->print();
    v2->scale(0.5);
    v2->print();
}

```

(א) 5 נקודות) בפקודה `v2->scale(0.5);` רשום איזו פונקציה תופעל (הפונקציה המוגדרת במחלקה `vector2` או במחלקה `complex`)? הסבר בקצרה.

(ב) 10 נקודות) מה התוכנית תדפיס?

(ג) 5 נקודות) מה יקרה אם נוריד את המילה `virtual` מהגדרת הפונקציה `print` (בשתי המחלקות)? מה יהיה הפלט במקרה זה? הסבר מדוע.

שאלה מספר 6 (20 נקודות)

נתונה התוכנית:

```

#include <iostream.h>
#define S 20
void swap(char &c1, char &c2){
    char tmp=c1;
    c1=c2;
    c2=tmp;
}
void what(char str[]){
    int i=1,j;
    for (; i<S && str[i]; ++i)
        for(j=i; j>0; --j)
            if(str[j]<str[j-1])
                swap(str[j], str[j-1]);
}
void main(){
    char str[S]="abccbabac";
    what(str);
    cout<<str;
};

```

- א) (3 נקודות) הסבר מדוע בפונקציה swap הערכים נשלחים כ reference (char&). כיצד היינו מממשים פונקציה זו ללא reference ?
- ב) (7 נקודות) מה מבצעת הפונקציה what
- ג) (7 נקודות) מה התוכנית תדפיס
- ד) (3 נקודות) מה משמעות התנאי $i < S \ \&\& \ str[i]$? מה הערך המקסימלי ש- i מקבל בתוכניתנו?

חלק ב': VHDL

שאלה מספר 7 (20 נקודות)

נתונה התוכנית הבאה:

- א. (6 נקודות) יש לכתוב עירור (TestBench) לתוכנית זו, עיני רכיב עצמאי. כלומר, רכיב שתפקידו לייצר עירור מלא לתוכנית. אין להראות את החייוטים/חיבורים להפעלת התוכנית הנבדקת.
- ב. (8 נקודות) יש לכתוב תוכנית זחה פונקציונלית, עיני תהליך (Process) אחד מדורבן שעון ועוד פקודות כרצונך.
- ג. (6 נקודות) יש לצייר את החומרה שתיווצר מהתוכנית הנתונה ולצרף הסברים מתאימים לתרשים. אין להראות את החיבורים ל- clk ול- rst, כדי לחסוך בקווים. להקפיד על תרשים ברור!

```
library ieee;
use ieee.std_logic_1164.all;
entity dff is
    port (clk, rst: in std_logic;
          d: in std_logic;
          q: out std_logic);
end;
architecture rtl of dff is
begin
    process (clk, rst)
    begin
        if rst='1' then q<='0';
        elsif clk'event and clk = '1' then
            q<=d;
        end if;
    end process;
end;

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity dffs_xors is
    GENERIC (Size: INTEGER:= 3);
    port (d, clk, rst: in std_logic;
          q: out std_logic_vector (Size-1 downto 0));
end;
architecture for_gen of dffs_xors is
    COMPONENT dff
        port (d, clk, rst: in std_logic; q :out std_logic);
    end component;
    -----
    SIGNAL qSig: std_logic_vector (Size-1 DOWNT0 0):= (others=>'0');
    SIGNAL dSig: std_logic_vector (Size-1 DOWNT0 0):= (others=>'0');
begin
    -----
    dSig(0)<=d;
    -----
    out_loop: FOR i in 0 to size-1 GENERATE
        all_ffs: dff port map(d=>dSig(i), clk=>clk, rst=>rst,q=>qSig(i));
    end GENERATE out_loop;
    -----
    xors: FOR i IN 1 TO size-1 GENERATE
        dsig(i)<=qsig(i-1) XOR dsig(i-1);
    END GENERATE xors;
    -----
    q<=qSig; end;
```

שאלה מספר 8 (20 נקודות)

נתונה התוכנית הבאה, כאשר בכל הסעיפים נתון `.input = "00001000"` יש להשלים את הטבלאות להלן וגם לצייר את צורות הגלים של הפלט `.div_by`

```
library ieee;
use ieee.STD_LOGIC_1164.all;
use ieee.STD_LOGIC_ARITH.all;
use ieee.STD_LOGIC_UNSIGNED.all;
entity div is
    generic (size: NATURAL:= 8);
    port (clk: in BIT;
          rst: in BIT;
          input: in std_logic_vector(size-1 downto 0);
          div_by: out std_logic);
end div;
architecture div of div is
    signal count : std_logic_vector(size-1 downto 0);
begin
    process (rst, clk)
    begin
        if rst = '1' then
            count<=(others=>'0');
            div_by<='0';
        elsif clk'event and clk = '1' then
            if count = input - 1 then
                count<=(others=>'0');
                div_by<='0';
            else
                if count > SHR(input,"01") -1 then
                    div_by<='0';
                else
                    div_by<='1';
                end if;
                count<=count+1;
            end if;
        end if;
    end process;
end;
```

א. (8 נקודות) עבור התוכנית לעיל כמו שהיא, יש למלא את הטבלה הבאה ע"י רישום ערכו של `:div_by`

count	div_by
0	
1	
2	
3	
4	
5	
6	
7	
0	

ב. (6 נקודות) יש למלא את הטבלה הבאה עבור המקרה הבא: נשנה את השורה המודגשת בתוכנית, ובמקומה נרשום את השורה הבאה:

`if count < conv_integer(input)/2 then`

count	div_by
0	
1	
2	
3	
4	
5	
6	
7	
0	

ג. (6 נקודות) יש למלא את הטבלה הבאה עבור המקרה הבא: נשנה את השורה המודגשת בתוכנית, ובמקומה נרשום את השורה הבאה:

`if count <= conv_integer(input)/2 -1 then`

count	div_by
0	
1	
2	
3	
4	
5	
6	
7	
0	

הערות:

- הפונקציה SHR קיימת בספריות המצורפות לתוכנית ומבצעת הזזה ימנית.
- הפונקציה conv_integer קיימת בספריות המצורפות לתוכנית ומתרגמת מווקטור ל-integer. ניתן להשתמש בחילוק ב-2 עבור הסוג integer.

שאלה מספר 9 (20 נקודות)

נתונה התוכנית הבאה:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity up_down_dir3 is
    generic(n: natural:=4);
    port (clk: in BIT; rst: in BIT;
          low_in, high_in: in STD_LOGIC_vector(n-1 downto 0);
          output: out std_logic_vector(n-1 downto 0));
end up_down_dir3;
architecture up_down_dir3 of up_down_dir3 is
    signal output_sig: std_logic_vector(n-1 downto 0);
    signal dir: BIT;
begin
    direction: process (clk, rst)
    begin
        if rst = '1' then
            dir<='0';
        elsif CLK'event and CLK = '1' then
            if output_sig = high_in then
                dir<='1';
            elsif output_sig = low_in then
                dir<='0';
            end if;
        end if;
    end process;
    up_down: process (clk)
    begin
        IF CLK'event and CLK = '1' then
            IF rst='1' then output_sig<=high_in;
            ELSE
                IF dir='0' THEN
                    if output_sig /= high_in then
                        output_sig<= output_sig + 1;
                    end if;
                ELSE
                    if output_sig /= low_in then
                        output_sig<= output_sig - 1;
                    end if;
                END IF;
            END IF;
        END IF;
    end process;
    output<= output_sig;
end up_down_dir3;

```

נתונים:

`low_in = 0, high_in = 5`

- א. (6 נקודות) יש לרשום את הערכים העשרוניים של output עבור 10 מחזורי שעון ללא איפוס (כלומר, שניתן 'rst=1' בשעון כלשהו, ולאחריו 10 מחזורי שעון כשה-rst באפס.
- ב. (6 נקודות) יש לרשום את ערכי dir לכל ערך של output כפי שמופיע בסעיף א'. יש לצייר טבלה המסכמת את סעיפים א', ב'.
- ג. (8 נקודות) יש לכתוב תהליך/תהליכים, ליצירת עירור (TestBench) לתוכנית לעיל. העירור צריך להכיל שלשה זוגות ערכים לכניסות low_in, high_in, ועוד עירור מתאים לכניסות clk, rst. בסעיף זה אין לכתוב עירור מלא הכולל ממשק וחיווטים, רק את התהליך/תהליכים המייצרים את העירור.

שאלה מספר 10 (20 נקודות)

נתונה הגדרת הפרוצדורה, בחבילה הבאה:

```

library ieee ;
use ieee.STD_LOGIC_1164.all;
use ieee.STD_LOGIC_ARITH.all;
use ieee.STD_LOGIC_UNSIGNED.all;
package pack is
    procedure cnt(vec: in std_logic_vector;
                  ones: out std_logic_vector);
end;
```

א. (8 נקודות) יש לכתוב פרוצדורה בשם cnt, בתוך package body, שתספור את מספר האחדים בווקטור vec, שאורכו איננו ידוע. את מספר האחדים יש להכניס לווקטור ones שאורכו זהה לאורך vec.

ב. (6 נקודות) יש לכתוב תוכנית פשוטה הקוראת לפרוצדורה הנ"ל ומשתמשת בה לספירת האחדים בווקטור הכניסה vec_in. להלן הממשק:

```

library ieee; use ieee.STD_LOGIC_1164.all;
use ieee.STD_LOGIC_ARITH.all;
use ieee.STD_LOGIC_UNSIGNED.all;
use work.pack.all;
entity test_procedure is
    generic(size: integer:=4);
    port(vec_in: in std_logic_vector(size-1 downto 0);
          ones_out: out std_logic_vector(size-1 downto 0));
end;
```

ג. (6 נקודות) יש להשלים את התוכנית הבאה (במקומות המסומנים במלבן) לספירת מספר האחדים בווקטור כניסה בן 4 סיביות (6 נק). מומלץ להוסיף תרשים להבהרת הפתרון!

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
PACKAGE pkg IS
    COMPONENT Ha
        PORT( a, b: IN STD_LOGIC;
              sum, cout: OUT STD_LOGIC);
    END COMPONENT;
    COMPONENT Fa
        PORT(a,b,c: IN STD_LOGIC;
              sum, cout: OUT STD_LOGIC);
    END COMPONENT;
END pkg;
-----
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY Ha IS
    PORT( a, b : IN STD_LOGIC;
          sum, cout :OUT STD_LOGIC);
END ha;
ARCHITECTURE ha OF Ha IS
BEGIN
    sum <= a XOR b;
    cout<= a AND b;
END ha;
-----
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY Fa IS
    PORT( a, b,c :IN STD_LOGIC;
          sum, cout :OUT STD_LOGIC);
END fa;
ARCHITECTURE Fa OF Fa IS
BEGIN
    sum <= a XOR b xor c;
    cout<= (a AND b) or (a AND c) or (b AND c);
END fa;
-----
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE WORK.PKG.ALL;
ENTITY cnt IS
    PORT(vec: IN STD_LOGIC_vector(3 downto 0);
          ones: OUT STD_LOGIC_vector(2 downto 0));
END;

ARCHITECTURE struct OF cnt IS
    SIGNAL sig: STD_LOGIC_vector(4 downto 0);
BEGIN
    box1: Ha port map (a=>          ,b=>          ,sum=>          ,cout=>          );
    box2: Ha port map (a=>          ,b=>          ,sum=>          ,cout=>          );
    box3: Ha port map (a=>          ,b=>          ,sum=>          ,cout=>          );
    box4: Fa port map (a=>          ,b=>          ,c=>          ,sum=>          ,
                                cout=>          );
END;

```