

מדעי המחשב – 2 יחידות לימוד
פתרון בחינת הבגרות

פרק ראשון

שאלה 1

```
public Student(string name, int year)
{
    this.name = name;
    this.year = year;
}
```

א.

```
public static void Main(string[] args)
{
    Student st1 = new Student("Aaa", 2000);
    Student st2 = new Student("Bbb", 2000);
}
```

ב.

שאלה 2

א.

	0	1	2	3	4	5	6	7	length
arr	12 0	2	324	33	67 0	888	9	5	8

i	i<8	arr[i]	arr[i]>9 && arr[i]<100
0	כן	12	כן
2	כן	324	לא
4	כן	67	כן
6	כן	9	לא
8	לא		

ב.

	0	1	2	3	4	5	6	7	length
arr	1	20	3	40	5	50	101	70	8

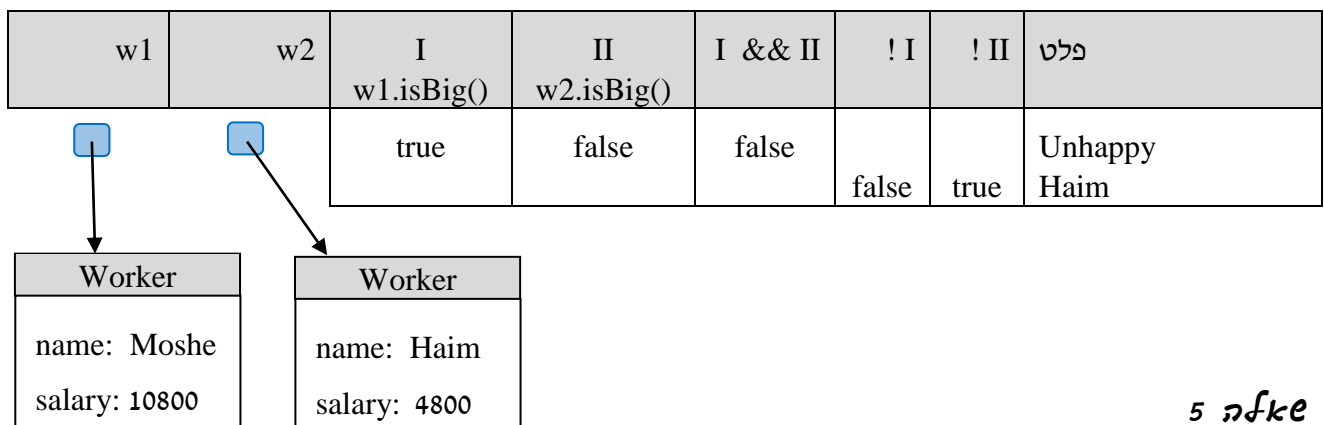
כל איברי המערך שנמצאים במקומות הזוגיים במערך קטנים מ- 10 או גדולים מ- 99

עלף 3

```
//--- פעולה המקבלת מספר שלם תלת ספרתי ומשתנה בוליאני ---
//--- אם המשתנה הבוליאני אמת - יוחזר סכום ספרות המספר ---
//--- אחרת - תוחזר מכפלת הספרות ---
public static int ChkNum(int num, bool isSum)
{
    int d1 = num % 10; // ספרת האחדות
    num = num / 10;
    int d10 = num % 10; // ספרת העשרות
    int d100 = num / 10; // ספרת המאות

    if (isSum)
        return d1 + d10 + d100;
    return d1 * d10 * d100;
}
```

עלף 4



עלף 5

.א

```
//--- פעולה המחזירה אמת אם חייב ועד בבניין ושקר אחרת ---
//--- חייב ועד בבניין אם יש בו יותר מ-4 דירות ---
public bool NeedVaad()
{
    return this.numAps > 4;
}
```

.ב

```
int count = 0;
for (int i = 0 ; i < bld.Length ; i++)
{
    if (bld[i].NeedVaad())
        count ++;
}
Console.WriteLine("count = " + count);
```

פרק שני

שאלה 6

```
//--- פעולה המחזירה אמת אם הישוב ---
//--- מתאים לתנאי הסקר ושקר אחרת ---
public bool IsFit()
{
    return numOfSchools > 3 && popul > 5000;
}

public static void Main(string[] args)
{
    int count = 0; // מונה מספר הלא מתאימים לסקר
    for (int i = 0 ; i < 248 ; i++)
    {
        Console.Write("city name --> ");
        string name = Console.ReadLine();
        Console.Write("population size --> ");
        int popul = int.Parse(Console.ReadLine());
        Console.Write("num of schools --> ");
        int numOfSchools = int.Parse(Console.ReadLine());

        City city = new City (name, popul, numOfSchools);

        Console.WriteLine("name: " + city.GetName() +
            " Is Fitting ? " + city.IsFit());

        if (! city.IsFit())
            count ++;
    }
    Console.WriteLine("don't fit: " + count);
}
```

אלף 7

```

//--- פעולה בונה ---
public Pen(string made)
{
    this.color = "red";
    this.made = made;
    this.price = 10.0;
    this.weight = 25;
}

//--- פעולה המעדכנת את מחיר העט ---
public void SetPrice(double x)
{
    this.price = x;
}

//--- פעולה המחזירה אמת אם מחיר העטים שווה ושקר אחרת ---
public bool IsSamePrice(Pen other)
{
    return this.price == other.price;
}

```

ד. טבלת מעקב:

a1		a2		a3		! a1.IsSame Price(a3)	a2.IsSame Price(a1)	פלט
Pen		Pen		Pen				
color	red	color	red	color	black	true	false	No
made	aaa	made	bbb	made	ccc			
price	10.0 15.0	price	10.0	price	8.0			
weight	5.5	weight	25.0	weight	12.5			

a1: [color = red, made = aaa, price = 10.0, weight = 5.5]
a2: [color = red, made = bbb, price = 10.0, weight = 25.0]
a3: [color = black, made = ccc, price = 8.0, weight = 12.5]

a1: [color = red, made = aaa, price = 15.0, weight = 5.5]

NO

אלף 8

.א

```
--- פעולה המחזירה את המחיר שעל המשפחה לשלם עבור ההשתתפות בטיוול ---  
public int CalcPrice()  
{  
    int sum = 100;    // מחיר ההדרכה  
  
    if (this.num <= 3)  
        sum = sum + 30 * this.num;  
    else  
        if (this.num <= 5)  
            sum = sum + 28 * this.num;  
        else  
            sum = sum + 26 * this.num;  
  
    return sum;  
}
```

אפשרות נוספת: שימוש ב-switch-case

```
--- פעולה המחזירה את המחיר שעל המשפחה לשלם עבור ההשתתפות בטיוול ---  
public int CalcPrice()  
{  
    int sum = 100;    // מחיר ההדרכה  
  
    switch (this.num)  
    {  
        case 1:  
        case 2:  
        case 3: sum = sum + 30 * this.num; break;  
        case 4:  
        case 5: sum = sum + 28 * this.num; break;  
        default: sum = sum + 26 * this.num; break;  
    }  
  
    return sum;  
}
```

```
public static void Main(string[] args)
{
    int count = 0, max = 0;

    Console.Write("num of participants --> ");
    int num = int.Parse(Console.ReadLine());
    while (num != 0)
    {
        Console.Write("family name? --> ");
        string name = Console.ReadLine();

        Family family = new Family (num, name);
        int toPay = family.CalcPrice();

        Console.WriteLine("name: " + family.GetFamilyName() +
            ", toPay: " + toPay);
        count += family.GetNum();
        if (toPay > max)
            max = toPay;

        Console.Write("num of participants --> ");
        num = int.Parse(Console.ReadLine());
    }

    Console.WriteLine("num of participants is: " + count);
    Console.WriteLine("max family price is: " + max);
}
```

פרק שלישי

אלף 9

```

//--- פעולה המחזירה את הקלף במקום p בחפיסה --- .א
public Card SeeCard(int i) { return this.allCard[i]; }

```

ב. הפתרון משתמש במחלקות הבאות:

```

/* חפיסת קלפים */
class Deck
{
    private Card[] allCard; // מערך הקלפים

    //--- פעולה המחזירה חפיסת קלפים מעורבבת ---
    public Deck()...

    //--- ערבוב החפיסה ---
    private void Suffle()...

    //--- פעולה המחזירה את הקלף במקום p בחפיסה ---
    public Card SeeCard(int i)...

    public override string ToString()...
}

/* מחלקה המגדירה שחקן */
class Player
{
    private int score; // מספר הנקודות לשחקן
    private Deck deck; // חפיסת הקלפים

    //--- פעולה בונה ---
    public Player()...

    //--- פעולה המחזירה קלף מהחפיסה של השחקן הנוכחי ---
    public Card SeeCard(int i){ return this.deck.SeeCard(i); }

    //--- פעולה המוסיפה נקודות לשחקן הנוכחי ---
    public void AddPoints(int points)...

    //--- פעולה המחזירה את מספר הנקודות שצבר השחקן ---
    public int GetScore()...

    public override string ToString()...
}

```

קטע התכנית המדמה את המשחק ומציג את מספר הנקודות שצבר כל שחקן בסיום המשחק

```
Player p1 = new Player();
Player p2 = new Player();

Console.WriteLine("p1: " + p1);
Console.WriteLine();
Console.WriteLine("p2: " + p2);
Console.WriteLine();

Random rnd = new Random();

int i;
while (p1.GetScore() < 28 && p2.GetScore() < 28)
{
    i = rnd.Next(36);
    Card c1 = p1.SeeCard(i);
    i = rnd.Next(36);
    Card c2 = p2.SeeCard(i);

    Console.WriteLine("p1: " + c1 + "\t p2: " + c2);

    int val = c1.CompareVal(c2);
    if (val == 1)
        p1.AddPoints(3);
    else
        if (val == 2)
            p2.AddPoints(3);
        else
        {
            p1.AddPoints(1);
            p2.AddPoints(1);
        }

    //--- למי יש צבע ירוק ? (3) ---
    if (c1.GetColor() == 3)
        p1.AddPoints(2);
    if (c2.GetColor() == 3)
        p2.AddPoints(2);

    Console.WriteLine("\t p1: " + p1.GetScore() + "\t p2: " + p2.GetScore());
}

Console.WriteLine("p1 got " + p1.GetScore() + " points");
Console.WriteLine("p2 got " + p2.GetScore() + " points");
```



```

//--- פעולה המחזירה חפיסת קלפים מעורבבת ---
public Deck()
{
    this.allCard = new Card[36];

    //--- מילוי החפיסה ---
    int p = 0;
    for (int i = 1; i <= 9; i++)
        for (int j = 1; j <= 4; j++)
            {
                allCard[p] = new Card(i, j);
                p++;
            }

    this.Suffle(); // ערבוב החפיסה
}

//--- ערבוב החפיסה ---
private void Suffle()
{
    Random rnd = new Random();

    for (int i = 0; i < 200; i++)
    {
        int x = rnd.Next(36);
        int y = rnd.Next(36);

        Card temp = allCard[x];
        allCard[x] = allCard[y];
        allCard[y] = temp;
    }
}
/*
p1: score: 0
[9, yellow] [6, blue] [7, yellow] [2, green] [3, green] [8, blue] [3, blue] [8, yellow] [1, blue]
[2, red] [8, green] [6, red] [9, blue] [1, yellow] [6, yellow] [4, green] [7, green] [9, green]
[9, red] [1, red] [3, yellow] [2, blue] [4, yellow] [4, blue] [5, yellow] [4, red] [5, green]
[7, blue] [5, red] [8, red] [7, red] [1, green] [3, red] [2, yellow] [6, green] [5, blue]

p2: score: 0
[9, yellow] [9, red] [3, red] [5, yellow] [1, blue] [6, blue] [5, blue] [1, green] [4, blue]
[7, green] [8, red] [5, red] [3, yellow] [3, blue] [1, red] [2, yellow] [4, green] [1, yellow]
[6, green] [4, red] [6, yellow] [7, yellow] [2, blue] [3, green] [4, yellow] [5, green] [8, yellow]
[8, blue] [8, green] [9, green] [6, red] [7, blue] [9, blue] [2, red] [2, green] [7, red]

p1: [1, red] p2: [2, yellow] p1: 0 p2: 3
p1: [6, blue] p2: [1, yellow] p1: 3 p2: 3
p1: [8, yellow] p2: [4, green] p1: 6 p2: 5
p1: [9, green] p2: [4, yellow] p1: 11 p2: 5
p1: [6, blue] p2: [4, red] p1: 14 p2: 5
p1: [3, yellow] p2: [3, yellow] p1: 15 p2: 6
p1: [2, blue] p2: [1, green] p1: 18 p2: 8
p1: [5, yellow] p2: [4, yellow] p1: 21 p2: 8
p1: [6, yellow] p2: [6, blue] p1: 22 p2: 9
p1: [8, blue] p2: [1, blue] p1: 25 p2: 9
p1: [9, yellow] p2: [5, red] p1: 28 p2: 9

p1 got 28 points
p2 got 9 points
*/

```

10 אלה

```
class Item .א
{
    private int value; // ערך האיבר
    private int row; // שורה במטריצה
    private int col; // עמודה במטריצה

    //--- פעולה בונה ---
    public Item(int value, int row, int col)
    {
        this.value = value;
        this.row = row;
        this.col = col;
    }
}

class Sparse .ב
{
    private Item[] itemAr; // מערך האיברים
    private int rows; // מספר השורות
    private int cols; // מספר העמודות

    public Sparse(int[,] mat) .ג
    {
        this.rows = mat.GetLength(0);
        this.cols = mat.GetLength(1);

        int size = CountNoZero(mat);
        this.itemAr = new Item[size];

        //--- מילוי המערך באיברי המטריצה הדלילה ---
        int p = 0;
        for (int i = 0; i < mat.GetLength(0); i++)
        {
            for (int j = 0; j < mat.GetLength(1); j++)
                if (mat[i,j] != 0)
                {
                    this.itemAr[p] = new Item(mat[i,j], i, j);
                    p++;
                }
        }
    }
}
```

תכנית היוצרת מטריצה של מספרים שלמים, ויוצרת מטריצה דלילה מתאימה: (לא נדרש בבחינה)

```
public static void Main(string[] args)
{
    int [,] mat = { {2, 0, 0, 3, 0},
                   {0, 0, 7, 0, 0},
                   {0, 9, 6, 0, 0},
                   {1, 0, 0, 0, 4}};

    Sparse sp = new Sparse(mat);

    int r = mat.GetLength(0);
    int c = mat.GetLength(1);
    for (int i = 0 ; i < r ; i++)
    {
        for (int j = 0 ; j < c ; j++)
            Console.Write(mat[i,j] + "\t");
        Console.WriteLine();
    }
    Console.WriteLine();

    Console.WriteLine(sp);
}

/*
2  0  0  3  0
0  0  7  0  0
0  9  6  0  0
1  0  0  0  4

0: [value = 2, (r.0, c.0)]
1: [value = 3, (r.0, c.3)]
2: [value = 7, (r.1, c.2)]
3: [value = 9, (r.2, c.1)]
4: [value = 6, (r.2, c.2)]
5: [value = 1, (r.3, c.0)]
6: [value = 4, (r.3, c.4)]
*/
```