I. Duota klasė *Program* (žr. žemiau). Kas bus atspausdinta ekrane, *Main* metode įvykdžius nurodytus veiksmus?

class Program

{

enum Ivertinimai {

Nevertinama,

LBlogai,

Blogai,

Nepatenkinamai3,

Nepatenkinamai4,

Silpnai,

Patenkinamai,

Vidutiniskai,

Gerai,

LGerai,

Puikiai

};

struct S

        {

            private int a;

            private int b;

            private int c;

            public S(int x) { a = b = c = x; } // ???

            public int  Imti() { return a + b + c; }

        };

static void Main(string[] args)

{

int x = 10;

for (Ivertinimai i = Ivertinimai.LGerai;

i >= Ivertinimai.LBlogai; i--)

{

x = (int)i;

Console.Write("{0} ", x - 1);

}

S s1 = new S();

            S s2 = new S(2);

            Console.WriteLine("Ats {0} {1}", s1.Imti(), s2.Imti());

}

}

Pasirinkite vieną:

1. 9 8 7 6 5 4 3 2 1 0 Ats 6 0
2. 8 7 6 5 4 3 2 1 0 Ats 0 6
3. 9 8 7 6 5 4 3 2 1 Ats 0 0
4. Programoje yra klaidų

II. Turime prekių sąrašą: *prekės pavadinimas, kaina eurais, nuolaida procentais*. Pabaikite realizuoti užklotą operatorių \* taip, kad prekių sąrašas būtų surikiuotas **pagal 3 požymius**: kainą eurais (didėjančiai), nuolaidą procentais (didėjančiai) ir prekės pavadinimą abėcėlės tvarka.

class Preke

    {

        // Klasės savybės

        public string Pavadinimas { get; set; }

        public double Kaina { get; set; }

        public int Nuolaida { get; set; }

        // Užklotas operatorius

        public static bool operator \* (Preke p1, Preke p2)

        {

            // Palygina p1 ir p2 prekių pavadinimus

            int p = String.Compare(p1.Pavadinimas, p2.Pavadinimas, StringComparison.CurrentCulture);

            // !!! return sakinys rašomas čia

        }

    }

    class Prekes

    {

        public Preke[] P { get; set; }      // Prekių objektų rinkinys P

        public int n     { get; set; }      // Prekių skaičius n

        // Duomenų rikiavimo metodas

        public void Rikiuoti() {

            int i = 0; bool bk = true;

            while (bk) {

                bk = false;

                for (int j = n - 1; j > i; j--)

                    // Naudojamas užklotas operatorius \*

                    if (P[j] \* P[j - 1])  {

                        bk = true;

                        Preke p  = P[j];

                        P[j]     = P[j - 1];

                        P[j - 1] = p;

                    }

                i++;

            }

        }

    }

Pasirinkite vieną:

a)      return ((p1.Kaina < p2.Kaina) ||

  (p1.Kaina == p2.Kaina && p1.Nuolaida > p2.Nuolaida) ||  
  (p1.Kaina == p2.Kaina || p1.Nuolaida == p2.Nuolaida && p > 0));

b)      return ((p1.Kaina < p2.Kaina) ||

  (p1.Kaina == p2.Kaina && p1.Nuolaida < p2.Nuolaida) ||  
  (p1.Kaina == p2.Kaina && p1.Nuolaida == p2.Nuolaida && p < 0));

c)      return ((p1.Kaina < p2.Kaina) &&

  (p1.Kaina == p2.Kaina || p1.Nuolaida < p2.Nuolaida) &&  
        (p1.Kaina == p2.Kaina || p1.Nuolaida == p2.Nuolaida || p < 0));

d)      return ((p1.Kaina < p2.Kaina) ||

  (p1.Kaina == p2.Kaina && p1.Nuolaida > p2.Nuolaida) ||  
  (p1.Kaina == p2.Kaina && p1.Nuolaida == p2.Nuolaida && p > 0));

III. 1. Duotos klasės *Elementas*, *Konteineris*. Klasių metodai, perdengti operatoriai yra realizuoti. Kokios bus konteinerinės klasės elementų objektų rinkinio A reikšmės *Program* klasės *Main* metode (žr. žemiau) įvykdžius nurodytus veiksmus?

    class Elementas

    {

        public int sk { get; set;}

        public Elementas(int sk = 0) { this.sk = sk; }    // Konstruktorius

        // Užkloti operatoriai

        public static bool operator ==(Elementas e1, Elementas e2) { return e1.sk == e2.sk; }

        public static bool operator !=(Elementas e1, Elementas e2) { return !(e1 == e2); }

    }

   class Konteineris

    {

        private Elementas[] A;         // Elementų objektų rinkinys A

        public int n { get; set; }     // Elementų kiekis n

        // Konstruktorius

        public Konteineris() { n = 0;  A = new Elementas[100]; }

        // Sąsajos metodai

        public Elementas Imti(int i) { return A[i]; }

        public void Deti(Elementas e) { A[n] = e; n++; }

        public void Metodas(int k, Elementas r)

        {

            if (n < 100) {

               for (int i = n; i > k; i--)

                 A[i] = A[i - 1];

               A[k] = r;

               n++;

            }

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            Konteineris K = new Konteineris();

            // !!! Konteinerio elementų užpildymas skaičiais

            // !!! A masyvo elementų reikšmės: 8 4 5 8 8 0 8 9

              Elementas ex = new Elementas(K.Imti(K.n / 2).sk);

            for (int j = 0; j < K.n; j++)

            {

                Elementas ej = K.Imti(j);

                if (ej == ex) {

                   Elementas ez = new Elementas(j);

                   K.Metodas(j+1, ez);

                }

            }

           // !!! Konteinerio elementų objektų rinkinio A spausdinimas atliekamas čia

        }

     }

Pasirinkite vieną:

1. 8 0 4 5 8 8 0 8 9
2. 8 0 4 5 8 4 8 6 0 8 9 9
3. 8 1 4 5 8 5 8 7 0 8 10 9
4. 8 0 4 5 8 1 8 2 0 8 3 9
5. Programa lūžta veikimo metu

IV. Kas bus atspausdinta ekrane, *Main* metode įvykdžius nurodytus veiksmus?

TimeSpan t = new TimeSpan(); // ciklo parametras

TimeSpan intervalas = new TimeSpan(0, 10, 0); // ciklo žingsnis

TimeSpan darboPradzia = new TimeSpan(8, 0, 0);

TimeSpan darboPabaiga = new TimeSpan(8, 45, 0);

DateTime data3 = new DateTime(2017, 10, 03, 16, 0, 0);

for (t = darboPradzia; t <= darboPabaiga; t = t + intervalas)

data3 = data3.AddMinutes(8.0);

Console.WriteLine(String.Format("{0:y MM ddd hhh mm} {0:yyy MMM d HHH mmm}",  
 data3, data3));

17 10 An 04 40 2017 Spl 16 40

17 10 An 04 40 2017 spalio 16 40

17 10 An 16 40 2017 Spl 16 40

2017 10 An 04 40 2017 Spl 16 40

17 10 An 04 48 2017 Spl 16 48

Teisingo atsakymo nėra