IE Info

Gongolie Catalin
SCAN 61

19,75

Ex.1 Linear ~~po~~ interpolation

Q1.1

```
public static double[][] transposeMatrix (double[][] a){
    double[][] b = new double[~~a.length[0]~~ a[0].length][a.length];
    for (int i=0; i< a.length; i++){
        for (int j=0; j< ~~a.length[i]~~ a[i].length; j++){
            b[j][i] = a[i][j];
        }
    }
    return b;
}
```

1

Q 1.2

```
public static void displayMatrix (double[][] a){
    for (int i=0; i<a.length; i++){
        for (int j=0; j<a[i].length; j++){
            System.out.print(a[i][j] + " ");
        }
        System.out.println();
    }
}
```

1

Q. 1.3

```
public static double[] matrixVectorProduct(double[][] a, double[] b){
    double[] c = new double[a.length];
    for (int i=0; i<a.length; i++){
        for (int j=0; j<a[i].length; j++){
            c[i] = c[i] + a[i][j] * b[j];
        }
    }
    return c;
}
```

1,5

Q 1.4

```
public static double[][] matrixProduct(double[][] a, double[][] b){
    double[][] c = new double~~[][]~~[a.length][b[0].length];
    for (int i=0; i<a.length; i++){
        for (int j=0; j<b.length; j++){
            for (int k=0; k<b[j].length; k++){
                c[i][k] = c[i][k] + a[i][j] * b[j][k];
            }
        }
    }
    return c;
}
```

it is OK
but mathematically speaking
we prefer to switch

2

Q 1.5

```
public static double[] solveSystem (double[] x, double[] y){
    double[] s = new double[2];
    double[][] u = new double[x.length][2];
    for (int i=0; i< x.length; i++){
        u[i][0] = x[i];
        u[i][1] = ~~1;~~ 1.0;
    }
    double[] s1 = new double[2];
2,5 s1 = matrixProduct(invertMatrix(matrixProduct(transposeMatrix(u), u)),
                       ~~matrixProduct(transposeMatrix(u),~~
                       matrixVectorProduct(transposeMatrix(u), y));
    S[0] = S1[0];
    S[1] = S1[1];
    return S;
}
```

Q 1.6

```
public static double computeError(double[] x, double[] y, double[] s){
    double ~~e=0;~~ error = 0;
    for (int i=0; i< x.length; i++){
1,5     error = error + (y[i] - s[0]*x[i] - s[1])*(y[i]-s[0]*x[i] - s[1]);
    }
    return error;
}
```

```
Q. 1.7   public static ~~main~~ String main (String arg[]){
             double x = ~~new double [6]~~, {0.8, 2.2, 2.8, 4.2, 4.8, 6.2};
             double y = ~~new double [6]~~, {11.9, 21.7, 31.8, 42.0, 51.9, 61.9};
             double s = new double [2];
             s = solve System (x, y);
1,5          int error = compute Error (x, y, s);
             System.out.println(~~s[0]~~ "a =" + s[0]);
             System.out.println ("b=" + s[1]);
             System.out.println (" Error is " + error);
         }
```

2. The return of the Binary

```
Q.2.1 public class Binary{
          public int b0;
          public int b1;
          public Binary (int b0, int b1){
              this.b0 = b0;
              this.b1 = b1;
          }
3         public Binary(){
              this.b0 = 0;
              this.b1 = 0;
          }
          public String toString(){
              return b1 + "" + b0;
          }
          // file with name Binary.class, stored in C:/Users/Catalin/Desktop/Binary
      }
```

IE Info - ~~cont~~
2nd part.

Gongolie Catalin
SCAN 61

Q. 2.2

1

```
public class TestBinary{
    public static String mein(String args[]){
        ~~new~~ Binary u = new Binary(1,0);
        Binary v = new Binary(1,1);
        System.out.println(u);
        System.out.println(v);
    }
    // file is stored with name TestBinary.java, in C:/Users/Catalin/Desktop/Binary
}
```

Q. 2.3

Methods for Binary class:

1,75

```
public int inDecimal(){
    int a;
    a = b0 + 2*b1;
    return a;
}

public ~~int~~ boolean isEven(){
    boolean even = false;
    if (b~~1~~0==0){
        even=true;
    }
    return even;
}
```

Code in main:

```
System.out.println("The decimal form is " + u.inDecimal());
If (u.isEven() == true){
    System.out.println(~~u +~~ "The number is "even");
} else {
    System.out.println("The number is odd");
}
```

(Q.2.4)

```
public static Binary integerToBinary(int v){
    Binary u = new Binary();
    u.b0 = V%2;
    ~~u.b1 = (int) v:2~~; double tmp = V:2;
    u.b1 = (int) tmp;
    return u;
}
```

no interest!

1,5

Not the idea!

↳ a constructor

```
public Binary(int u)
{

}
```

(Q 2.5)

The addition of u and v are logically, by following the idea of finery and declaration of the class Binary, only if their some together don't exceed $(11)_2$ (or $(3)_{10}$), because otherwise they'll get out of the idea of binary computation. ~~The~~

The addition with 5 is already impossible because it exceeds the representation in 2 bits.

If we still suppose that we can loose ~~some~~ the bits that exceed the representation in 2 bits, then we'll get a wrong answer represented in base 10.

1,5

Implementation of addInteger method if it follows the instructions

```
    ~~public addInteger~~
  public ~~Binary addInteger( Binary v){~~
      ~~v.b0 = u.b0 + v.b0;~~
      ~~v.b1 = u.b1 + u.b1;~~
  }   ~~return v;~~
  }
  }
```

~~but still that's not possible to~~

```
public void addInteger( Binary v){
    v.b0 = this.b0 + v.b0;
    v.b1 = this.b1 + v.b1;
}
```