


| | | |
|--|--|---------------------|
| Programming June 2010 Leganés |  Universidad Carlos III de Madrid | |
| <i>Surname</i> | <i>Name</i> | <i>Group</i> |
| | | |

READ CAREFULLY THESE INSTRUCTIONS BEFORE STARTING THE EXAM:

- Fill in all the pages with a pen (personal data and answers)
- Do not use a pencil or a red pen
- Do not forget your NIA and your actual group
- The duration of this exam is 3 hours
- Books and notes are allowed
- Use only these sheets for the questions 1-3. Use the back of the pages if needed. Additional sheets will not be considered for this part
- Use the provided sheets for the problems 1-7.

DO NOT CONTINUE WITH THE NEXT PAGE until indicated

PART I: QUESTIONS

Question 1 (1 Point).- **Explain** the result of the method below (i.e., which is the returned value).

```
public static int [][] method1 (int [] val){
    int k=0, j=0;
    int result [][];
    while (k<val.length) {
        if (val[k]==1) j=j+1;
        k++;
    }
    k=0;
    result = new int [j][2];
    for (j=1;j<val.length;j++){
        if (val[j]==1) {
            result[k][0]=j-1;
            result[k][1]=val[j-1];
            k++;
        }
    }
    return result;
}
```

The result of this method is an two-dimension array of integers that stores the value and the position of the elements in val before an element with value '1'. The size of result is <number of 1s of val> x 2.

For example, if val = {3, 2, 1, 1, 2, 3, 1, 2, 1}, result = { {1, 2}, {2, 1}, {5, 3}, , {7, 2} }.

There is an exception: if the array val begins with a '1', the method does not take into account this value. In this case, the first pair of values of result (result[0][0], result[0][1]) correspond to the second '1' in val, and the last pair of values of result (result[result.length-1][0], result[result.length-1][1]) are equal to 0.

Question 2 (1 Point) .- **Find and explain** the 4 compilation errors included in the following Java code. **Explain** how they can be solved.

```
public class Question2 {
    static int c;

    public static void method1(String a, String b){
        String c = a+b;
        if (c.length()>2)
            c = c.substring(2);
        else c=b+a;
        System.out.println(c);
    }

    public static int method2 (){
        return c;
    }

    public static String method3(int j){
        c=j;
        System.out.println(c);
        return "";
    }

    public static void method4 (int j){
        int k = 0;
        c=k+j;
        System.out.println(c);
    }

    public static void main(String[] args) {
        int k = 16;
        method2();
        k = method2();
    }
}
```

Missing return type

Missing return instruction

k has not been initialized

method2 has been declared without parameters

Question 3 (1 Point).- **Explain the output** (i.e., what is printed on the screen) resulting from the execution of the `main` method below:

```
public class Question3 {  
    public static void main(String [] args) {  
        int [] N = new int [] {1, 2, 3, 4, 5};  
        int k = 2;  
  
        method1(N, k + 1);  
  
        System.out.println(k);  
        method2(N);  
    }  
  
    public static void method1(int [] N, int k) {  
        for(int i=0; i < N.length; i++) {  
            N[i] *= k;  
        }  
        k = 0;  
    }  
  
    public static void method2(int [] N) {  
        System.out.print("[");  
        for(int i=0; i<N.length; i++) {  
            System.out.print(N[i] + " ");  
        }  
        System.out.print("]");  
    }  
}
```

The program prints on the screen the array resulting from the scalar product of the integer value $k+1$ ($= 3$) and the array N ($=\{1, 2, 3, 4, 5\}$), which is the array $\{3, 6, 9, 12, 15\}$, and the value of k in the main method, which is 2. Thus, the output is:

```
2  
[3 6 9 12 15 ]
```

It is important to notice that the values of the array N are changed in the method `method1`. As it is known, the array reference itself cannot be changed, but the array values can be. Similarly, the changes to the local copy of $k+1$ value in `method1` does not have effect out of this method.

PART II: PROBLEMS

Problem 1 (1 Point).- Implement a method named `count` with two parameters: an array of `String` (named `A`) and a `String` (named `s`). The method must return the `int` value resulting from counting how many elements of the array `A` begin with the string `s`.

For example, if we execute:

```
String [] list = new String []  
    {"hello", "ciao", "hola", "hei", "salut", "moi"};  
int b = count(list, "he");
```

`b` value should be 2

Tip: The `String` class has a method named `startsWith` with the following specification:

```
public boolean startsWith(String prefix)  
Tests if the string starts with the specified prefix.
```

```
public static int count(String [] A, String s) {  
    int c = 0;  
    for(int i=0; i<A.length; i++)  
        if(A[i].startsWith(s))  
            c++;  
    return c;  
}
```

Problem 2 (1 Point).- Implement a method named `countSeveral` with two parameters: an array of `String` (named `A`) and an array of string (named `v`). The method must print on the screen how many strings of the array `A` begin with each string of the array `v`.

For example, if we execute:

```
String [] list = new String []  
    {"hello", "ciao", "hola", "hei", "salut", "moi"};  
String [] c = new String []  
    {"he", "h", "sal"};  
countSeveral(list, c);
```

The method must print:

```
2 words begin with he  
3 words begin with h  
1 words begin with sal
```

```
public static void countSeveral(String [] A, String [] S) {  
    for(int i=0; i<S.length; i++) {  
        int c = count(A, S[i]);  
        System.out.println(c + " words begin with " + S[i]);  
    }  
}
```

Problem 3 (1 Point).- Implement a method to sort arrays of `double` values in decreasing order with the *insertionsort* algorithm.

```

public static void insertionSort(double [] list) {

    for(int i=1; i < list.length; i++) {
        double e = list[i];
        int j = i-1;

        while(j>=0 && list[j] < e) {
            list[j+1] = list[j];
            j--;
        }

        list[j+1] = e;
    }
}

```

Problem 4 (1 Point).- Implement a method named `numbers` with no parameters. The method must return an `int` array with 30 random numbers in `[2, 20)`.

Tip: Remember the specification of the `random` method of the `Math` class:

```
public static double random()
```

Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

```

public static int [] numbers() {
    int [] n = new int[30];
    for(int i=0; i<n.length; i++)
        n[i] = 2 + (int) (Math.random() * 18);
    return n;
}

```

Problem 5 (1 Point).- Implement a method named `contains` with two parameters: a 2-dimension float array of any size (named `M`), and a float value (named `n`). The method must return a boolean value representing whether `n` is in `M` or not.

```

public static boolean contains(float [][] M, float n) {
    for(int i=0; i<M.length; i++)
        for(int j=0; j<M[0].length; j++)
            if(M[i][j] == n)
                return true;
    return false;
}

```

Problem 6 (1 Point).- Implement a method named `largest` with no parameters. The method must read positive integer numbers from the keyboard until -1 is introduced, and return the largest of the typed values.

```

public static int largest() {
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
}

```

```

    int v = 0;
    int max = -1;

    while(v != -1) {
        System.out.print("Enter a value (-1 to end): ");
        try {
            String s = br.readLine();
            v = Integer.parseInt(s);
        } catch(IOException e) {
            System.out.println("Error while reading.");
            System.exit(-1);
        } catch(NumberFormatException e) {
            System.out.println("Incorrect number format.");
            continue;
        }

        if(v < 0)
            break;
        else
            if(v > max)
                max = v;
    }

    return max;
}

```

Problem 7 (1 Point).- Implement a Java program (with a `main` method) that receives as arguments to the `main`: a double value (named `x`), and several integer values (named `coefficients`). The program must calculate the value of the polynomial expression defined by the `coefficients` for the value `x` and print it on the screen.

For example,

If the program is invoked with the following syntax: `java Problem7 1.8 4 2 3`

The polynomial expression defined by the coefficients is: $4 + 2x + 3x^2$

The result printed on the screen must be: 17.32 (since $4 + 2x1.8 + 3x1.8^2 = 17.32$).

The program must work for any value of `x` and any number of coefficients.

```

public class Problem7 {

    public static void main(String [] args) {

        // Check number of arguments
        if(args.length < 2) {
            System.out.println("[ERROR] Syntax: ");
            System.out.println(
                "java Problem7 <value x> <coefficient x^0> <coefficient x^1> ...");
            System.exit(-1);
        }
    }
}

```

```
// x
double x = 0;
try {
    x = Double.parseDouble(args[0]);
} catch (NumberFormatException e) {
    System.out.println(
"[ERROR] Argument <value x> is not a double value.");
    System.exit(-1);
}

// coefficients
int [] c = new int[args.length - 1];
try {
    for(int i=1; i<args.length; i++)
        c[i-1] = Integer.parseInt(args[i]);
} catch (NumberFormatException e) {
    System.out.println(
"[ERROR] One of the coefficients is not an integer value.");
    System.exit(-1);
}

/* Calculate polynomial expression value */
double r = 0;
for(int i=0; i<c.length; i++) {

    // term i
    double aux = 1;
    for(int j=0; j < i; j++) {
        aux = aux * x;
    }

    r = r + c[i] * aux;
}

/* Print result */
System.out.println("Result: " + r);
}
}
```


Programming – Grado en Ingeniería Informática

Authors

Of the English version:

Juan Gómez Romero

Based on the work by:

Ángel García Olaya

Manuel Pereira González

Silvia de Castro García

Gustavo Fernández-Baillo Cañas

Daniel Pérez Pinillos

Javier Ortiz Laguna

Álvaro Torralba Arias de Reyna



Universidad
Carlos III de Madrid