2021-02-16

ES 1

```
public static String maxSub ( String[] s)

/*  @ requires  s != null  && s.length > 0  && (\forall int i; i>=0 && i< s.length ; s[i] != null)
    @ ensures  (\forall int i; i>=0  && i< s.length ;
    @                 s[i]. contains (\result)) ;
    @ ensures  ! (\exists String str ;  str. length() > \result.length() ;  (\forall int i; i>=0 && i< s.length ;
    @                                                                          s[i] . contains ( str)) ;
    @ ensures  (\result == null) <==>  ! ( \exists  String str ; str!= null && str. length() > 0 ;
  */                                                                (\forall int i; i>=0 && i< s. length ;  s[i]. contains(s))) ;
```

ES 2

```
(A) public static int extract() throws ExceptionTombola {}

    /*  ...
      @ ensures  ! this. numeriEstratti(). contains (\result) && \result > 0 && result <= 90
      @ ensures  (\forall int i;  \old. numeriEstratti(). contains (i) ; this. numeriEstratti(). contains (i) ) &&
      @          ( this. numeriEstratti(). size() =  \old. numeriEstratti(). size() + 1) &&
      @          (this. numeriEstratti().get( \old. numeriEstratti. size()) == \result) ;
      @ ensures  ! (\exists Cartella c ;  c!=null && this. cartelle (). contains (c) ;  this. tombola (c) && ! old. tombola (c)) ;
      @
    */
```

```
(B) /*@ public invariant  (\forall Cartella c ;  this. cartelle(). contains (c) && c!= null ;
     @                         ! (\exist Cartella d ;  this. cartelle(). contains (d) && d != null && d != c ;
     @                              (\forall numeri_c ;  c. numeri. contains (numero_c) ;
     @                                   (\exists numero_d; d. numeri. contains(numero_d) ;  numero_d == numero_c))))
     @                  &&
     @                 (\ forall int i ; i>0 && i<=90 ;  (\exist Cartella c :  c!=null && this. cartelle(). contains (c) ;
     @                                                             c. numeri (). contains (i) )
    */
```

ES 3

```
public class Pista {
    private int num ;
    private int cp ;  // numero attuale di piloti

    public Pista (int num)  { this. num = num ; }

    public synchronized boolean enter () {
          if ( cp < num) { cp++ ; return true ; }
          (this. getSlowerPilot()) . exit () ;
          return false ;         ↳ restituisce il pilota
    }                                   più lento
    public synchronized exit() {
          cp -- ;
          notifyAll();
    }
}
```

```
public class Pilota implements Runnable {
      private Pista p ;

      public Pilota ( Pista p) { this. p = p; }

      public void exit () { p. exit () ; this. interrupt()}

      @ Override
      public void run() {
            synchronized ( p) {
                 while ( ! p. enter ()) {
                       try { p. wait ; }
                       catch ( Exception e ) {} ;
                 }
            try { Thread . sleep (random)}
            catch ( Exception e) {}
            this. exit() ;
      }
}
```

ES4

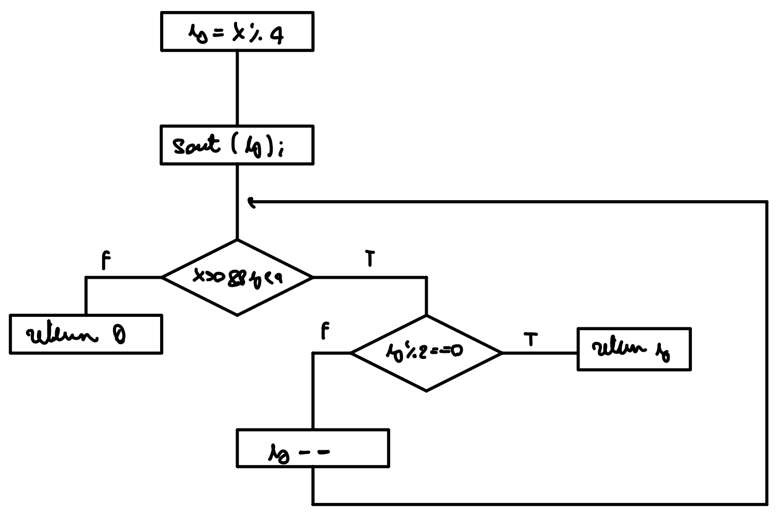
```
public class C {

    public C () {}

    public C ( int i) {
        this.i = i;
    }

    @ override
    public String toString () {
        return String.valueOf (i);
    }
}
```

ES5



- Istruzioni   I1 < 0 >
                I2 < 3 >
- Decisioni   D1 < 0 >
               D2 < 3 >
- Condizioni

x > 0	y<4	
0	0	→ $\cancel{1}$
0	1	→ <0
1	0	→ $\cancel{3}$
1	1	→ (3)