A drawing of a building

Description automatically generated

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

**ЛАБОРАТОРНА РОБОТА №6**

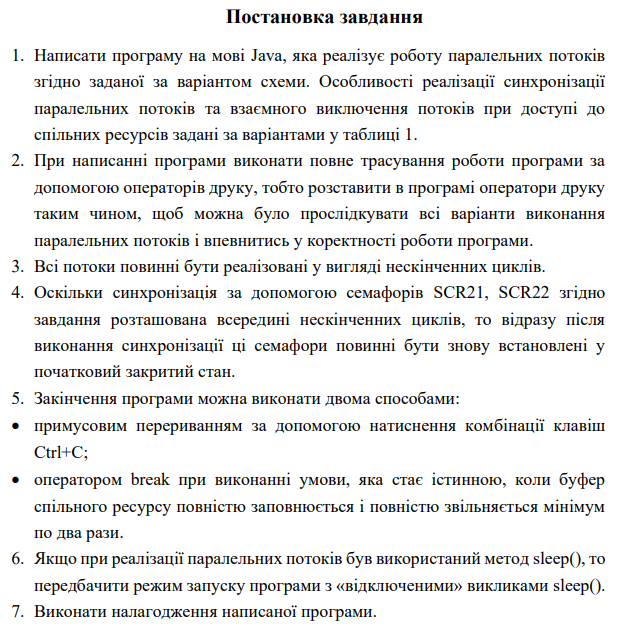
**КОМПЛЕКСНЕ ВИКОРИСТАННЯ ЗАСОБІВ ВЗАЄМОДІЇ**

**ПАРАЛЕЛЬНИХ ПОТОКІВ МОВИ JAVA**

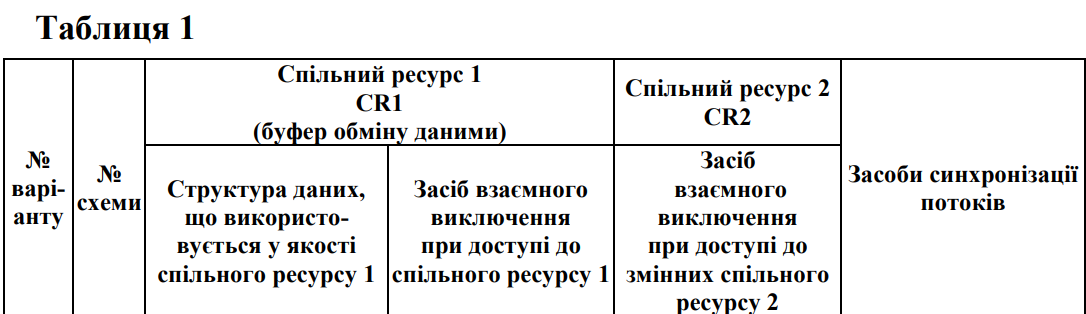
Виконав студент групи: КВ-22

ПІБ: Землянський Едуард

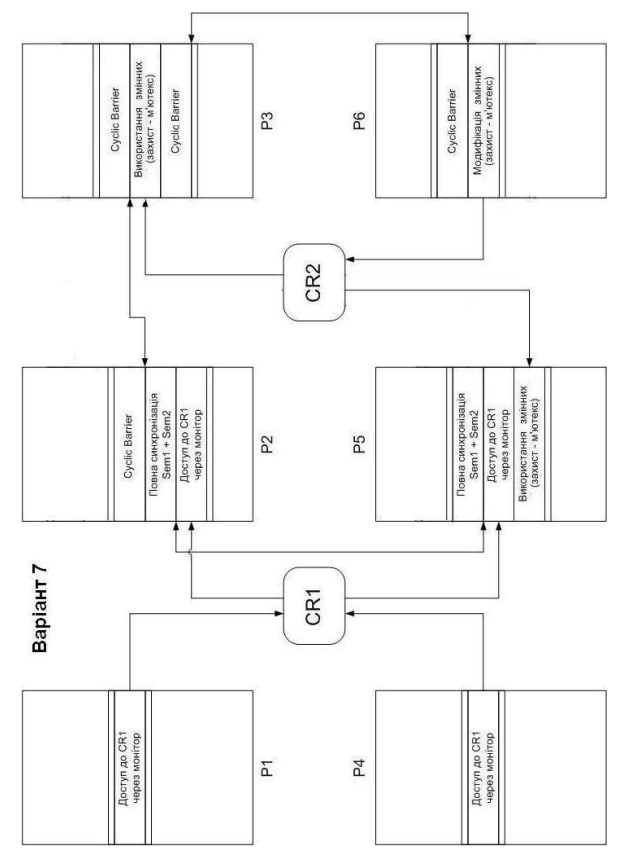
**Київ 2024**

****

**Варіант**

****

****



**Код:**

// № варіанту 7

// № схеми 7

// Спільний ресурс 1 CR1 (буфер обміну даними):

// Структура даних, що використовується у якості спільного ресурсу 1 - Стек у вигляді вектора

// Засіб взаємного виключення при доступі до спільного ресурсу 1 - Монітор

// Спільний ресурс 2 CR2:

// Засіб взаємного виключення при доступі до змінних спільного ресурсу 2 - М’ютекс

// Землянський Едуард КВ-22

import java.util.concurrent.CyclicBarrier; // for cyclic barrier

import java.util.concurrent.BrokenBarrierException;

import java.util.concurrent.Semaphore; // for semaphores

import java.util.Random;

import java.util.concurrent.locks.\*; // for mutex

class Global // sems for P2 and P5

{

    public static Random random = new Random();

    public static Semaphore thread\_sem1 = new Semaphore(0, true);

    public static Semaphore thread\_sem2 = new Semaphore(0, true);

    public static ReentrantLock mutex = new ReentrantLock();

}

class CR1

{

    public static final int MaxBufSize = 300;

    public static final int MinBufSize = 0;

    int buf[] = new int[MaxBufSize+1];

    int ind = 0;

    boolean IsEmpty = ind  == MinBufSize;

    boolean IsFull = ind == MaxBufSize;

    String Thread\_name;

    CR2 MyCR2 = new CR2();;

    synchronized void consume(String Thread\_name)

    {

        while (IsEmpty)

            try

            {

                wait();

            }

            catch (InterruptedException e)

            {

                System.out.println("InterruptedException");

            }

        System.out.println("Thread "+ Thread\_name + " consumed " + buf[ind] + " from buf[" + ind + "]");

        buf[ind] = 0;

        ind--;

        IsEmpty = ind == MinBufSize;

        IsFull = false;

        notify();

    }

    synchronized void produce (String Thread\_name)

    {

        while (IsFull){

            System.out.println("Producer " + Thread\_name + " terminated the programm !");

            MyCR2.print();

            System.exit(0);

        }

        ind++;

        buf[ind] = Global.random.nextInt(-100, 100);

        System.out.println("Thread " + Thread\_name + " produced " + buf[ind] + " into buf[" + ind + "]");

        IsFull = ind == MaxBufSize;

        IsEmpty = false;

        notify();

    }

}

class CR2 {

    // Global variables with starting 'zero' values

    public static byte byteVariable = 0;

    public static short shortVariable = 0;

    public static int intVariable = 0;

    public static long longVariable = 0L;

    public static float floatVariable = 0.0f;

    public static double doubleVariable = 0.0;

    public static boolean booleanVariable = false;

    public static char charVariable = ' ';

    // Method to access variable values

    void access(String Thread\_name) {

        System.out.println("Thread " + Thread\_name + " accessed CR2 data (used)");

    }

    // Method to print variable values

    public void print() {

        System.out.println("byteVariable: " + byteVariable);

        System.out.println("shortVariable: " + shortVariable);

        System.out.println("intVariable: " + intVariable);

        System.out.println("longVariable: " + longVariable);

        System.out.println("floatVariable: " + floatVariable);

        System.out.println("doubleVariable: " + doubleVariable);

        System.out.println("booleanVariable: " + booleanVariable);

        System.out.println("charVariable: " + charVariable);

    }

    // Method to edit variable values

    void edit (String Thread\_name){

        System.out.println("Thread " + Thread\_name + " accessed CR2 data (edited)");

        byteVariable = (byte) Global.random.nextInt(Byte.MAX\_VALUE + 1);

        shortVariable = (short) Global.random.nextInt(Short.MAX\_VALUE + 1);

        intVariable = Global.random.nextInt();

        longVariable = Global.random.nextLong();

        floatVariable = Global.random.nextFloat();

        doubleVariable = Global.random.nextDouble();

        booleanVariable = !booleanVariable;

        int randomAscii = Global.random.nextInt(26) + 97;

        charVariable = (char) randomAscii;

    }

}

class P1 implements Runnable // ready

{

    Thread t;

    CR1 CR;

    P1 (CR1 CR1\_arg)

    {

        this.CR = CR1\_arg;

        t = new Thread (this, "P1");

        t.start();

    }

    public void run()

    {

        while (true)

        {

            CR.produce(t.getName());

        }

    }

}

class P2 implements Runnable // ready

{

    Thread t;

    private CyclicBarrier br1;

    CR1 CR;

    P2 (CR1 CR\_arg, CyclicBarrier brInit)

    {

        this.CR = CR\_arg;

        t = new Thread (this, "P2");

        br1 = brInit;

        t.start();

    }

    public void run()

    {

        while (true)

        {

            // full sync with P3 via cyclic barr

            try{

                br1.await();

            }

            catch(BrokenBarrierException e){

                System.out.println(e.getMessage());

            }

            catch(InterruptedException e){

                System.out.println(e.getMessage());

            }

            // sem1 + sem2 full sync

            System.out.println("Thread P2 released semaphore for P5");

            Global.thread\_sem2.release();

            try{

                Global.thread\_sem1.acquire();

            }

            catch(InterruptedException e){

                System.out.println("Thread\_1 interrupted");

            }

            System.out.println("Thread P2 got semaphore released from P5");

            // acces to CR1 (taking)

            CR.consume(t.getName());

        }

    }

}

class P3 implements Runnable

{

    Thread t;

    private CyclicBarrier br1;

    private CyclicBarrier br2;

    CR2 CR;

    P3 (CR2 CR\_arg, CyclicBarrier brInit1, CyclicBarrier brInit2)

    {

        this.CR = CR\_arg;

        t  = new Thread (this, "P3");

        br1 = brInit1;

        br2 = brInit2;

        t.start();

    }

    public void run()

    {

        while (true)

        {

            // full sync with P2 via cyclic barr

            try{

                br1.await();

            }

            catch(BrokenBarrierException e){

                System.out.println(e.getMessage());

            }

            catch(InterruptedException e){

                System.out.println(e.getMessage());

            }

            // acessing CR2 with mutex

            Global.mutex.lock();

            CR.access(t.getName());

            Global.mutex.unlock();

            // full sync with P6 via cyclic barr

            try{

                br2.await();

            }

            catch(BrokenBarrierException e){

                System.out.println(e.getMessage());

            }

            catch(InterruptedException e){

                System.out.println(e.getMessage());

            }

        }

    }

}

class P4 implements Runnable

{

    Thread t;

    CR1 CR;

    P4 (CR1 CR\_arg)

    {

        this.CR = CR\_arg;

        t = new Thread (this, "P4");

        t.start();

    }

    public void run()

    {

        while (true)

        {

            // acces to CR1 (loading)

            CR.produce(t.getName());

        }

    }

}

class P5 implements Runnable

{

    Thread t;

    CR1 CR1;

    CR2 CR2;

    P5 (CR1 CR1\_arg, CR2 CR2\_arg)

    {

        this.CR1 = CR1\_arg;

        this.CR2 = CR2\_arg;

        t = new Thread (this, "P5");

        t.start();

    }

    public void run()

    {

        while (true)

        {

            // sem1 + sem2 full sync

            System.out.println("Thread P5 released semaphore for P2");

            Global.thread\_sem1.release();

            try{

                Global.thread\_sem2.acquire();

            }

            catch(InterruptedException e){

                System.out.println("Thread\_2 interrupted");

            }

            System.out.println("Thread P5 got semaphore released from P2");

            // acces to CR1 (taking)

            CR1.consume(t.getName());

            // acces to CR2 (using)

            Global.mutex.lock();

            CR2.access(t.getName());

            Global.mutex.unlock();

        }

    }

}

class P6 implements Runnable

{

    Thread t;

    private CyclicBarrier br2;

    CR2 CR;

    P6 (CR2 CR\_arg, CyclicBarrier brInit2)

    {

        this.CR = CR\_arg;

        t = new Thread (this, "P6");

        br2 = brInit2;

        t.start();

    }

    public void run()

    {

        while (true)

        {

            // full sync with P6 via cyclic barr

            try{

                br2.await();

            }

            catch(BrokenBarrierException e){

                 System.out.println(e.getMessage());

            }

            catch(InterruptedException e){

                System.out.println(e.getMessage());

            }

            // accesing CR2 (editing) with mutex

            Global.mutex.lock();

            CR.edit(t.getName());

            Global.mutex.unlock();

        }

    }

}

class Main {

    public static void main(String args[]) {

        CyclicBarrier br1 = new CyclicBarrier(2);

        CyclicBarrier br2 = new CyclicBarrier(2);

        // Creating an instance of CR1 and CR2

        CR1 MyCR1 = new CR1();

        CR2 MyCR2 = new CR2();

        MyCR2.print();

        new P1 (MyCR1);

        new P2 (MyCR1, br2);

        new P3 (MyCR2, br1, br2);

        new P4 (MyCR1);

        new P5 (MyCR1, MyCR2);

        new P6 (MyCR2, br2);

    }

}

**Результати виконання програми:**

byteVariable: 0

shortVariable: 0

intVariable: 0

longVariable: 0

floatVariable: 0.0

doubleVariable: 0.0

booleanVariable: false

charVariable:

Thread P5 released semaphore for P2

Thread P2 released semaphore for P5

Thread P5 got semaphore released from P2

Thread P2 got semaphore released from P5

Thread P6 accessed CR2 data (edited)

Thread P1 produced 5 into buf[1]

Thread P1 produced 4 into buf[2]

Thread P1 produced -100 into buf[3]

Thread P1 produced 70 into buf[4]

Thread P2 consumed 70 from buf[4]

Thread P2 released semaphore for P5

Thread P6 accessed CR2 data (edited)

Thread P5 consumed -100 from buf[3]

Thread P4 produced -31 into buf[3]

Thread P4 produced 94 into buf[4]

Thread P5 accessed CR2 data (used)

Thread P4 produced 38 into buf[5]

Thread P5 released semaphore for P2

Thread P4 produced 83 into buf[6]

Thread P2 got semaphore released from P5

Thread P5 got semaphore released from P2

Thread P4 produced -57 into buf[7]

Thread P4 produced -11 into buf[8]

Thread P4 produced 84 into buf[9]

Thread P4 produced -31 into buf[10]

Thread P4 produced 25 into buf[11]

Thread P4 produced -33 into buf[12]

Thread P1 produced 8 into buf[13]

Thread P1 produced 91 into buf[14]

Thread P1 produced -36 into buf[15]

Thread P1 produced 53 into buf[16]

Thread P1 produced 81 into buf[17]

Thread P1 produced -41 into buf[18]

Thread P1 produced -100 into buf[19]

Thread P1 produced 9 into buf[20]

Thread P1 produced 53 into buf[21]

Thread P1 produced -76 into buf[22]

Thread P1 produced 28 into buf[23]

Thread P1 produced 69 into buf[24]

Thread P1 produced -40 into buf[25]

Thread P1 produced -46 into buf[26]

Thread P1 produced 48 into buf[27]

Thread P1 produced 92 into buf[28]

Thread P1 produced 49 into buf[29]

Thread P1 produced 12 into buf[30]

Thread P1 produced -17 into buf[31]

Thread P1 produced 41 into buf[32]

Thread P1 produced 69 into buf[33]

Thread P1 produced -69 into buf[34]

Thread P1 produced 52 into buf[35]

Thread P1 produced -54 into buf[36]

Thread P1 produced 75 into buf[37]

Thread P1 produced 72 into buf[38]

Thread P1 produced 20 into buf[39]

Thread P1 produced 24 into buf[40]

Thread P1 produced 30 into buf[41]

Thread P4 produced 29 into buf[42]

Thread P4 produced -14 into buf[43]

Thread P5 consumed -14 from buf[43]

Thread P5 accessed CR2 data (used)

Thread P5 released semaphore for P2

Thread P2 consumed 29 from buf[42]

Thread P2 released semaphore for P5

Thread P6 accessed CR2 data (edited)

Thread P4 produced 78 into buf[42]

Thread P2 got semaphore released from P5

Thread P5 got semaphore released from P2

Thread P4 produced 16 into buf[43]

Thread P4 produced -97 into buf[44]

Thread P4 produced 86 into buf[45]

Thread P4 produced -78 into buf[46]

Thread P4 produced -60 into buf[47]

Thread P4 produced 21 into buf[48]

Thread P4 produced 43 into buf[49]

Thread P4 produced 75 into buf[50]

Thread P4 produced -94 into buf[51]

Thread P4 produced 55 into buf[52]

Thread P4 produced -43 into buf[53]

Thread P4 produced -84 into buf[54]

Thread P4 produced 89 into buf[55]

Thread P4 produced 80 into buf[56]

Thread P4 produced -98 into buf[57]

Thread P4 produced -8 into buf[58]

Thread P4 produced -82 into buf[59]

Thread P4 produced 37 into buf[60]

Thread P4 produced -58 into buf[61]

Thread P4 produced -63 into buf[62]

Thread P4 produced -80 into buf[63]

Thread P4 produced 19 into buf[64]

Thread P4 produced -5 into buf[65]

Thread P4 produced -100 into buf[66]

Thread P4 produced 51 into buf[67]

Thread P4 produced 44 into buf[68]

Thread P4 produced -54 into buf[69]

Thread P4 produced 34 into buf[70]

Thread P4 produced -41 into buf[71]

Thread P4 produced -44 into buf[72]

Thread P4 produced 63 into buf[73]

Thread P4 produced 8 into buf[74]

Thread P4 produced 69 into buf[75]

Thread P1 produced -97 into buf[76]

Thread P1 produced 12 into buf[77]

Thread P1 produced 64 into buf[78]

Thread P1 produced 54 into buf[79]

Thread P1 produced 77 into buf[80]

Thread P1 produced -86 into buf[81]

Thread P1 produced -48 into buf[82]

Thread P1 produced 87 into buf[83]

Thread P1 produced 29 into buf[84]

Thread P1 produced 98 into buf[85]

Thread P1 produced -87 into buf[86]

Thread P1 produced 20 into buf[87]

Thread P1 produced 71 into buf[88]

Thread P1 produced -66 into buf[89]

Thread P1 produced -46 into buf[90]

Thread P1 produced 88 into buf[91]

Thread P1 produced -2 into buf[92]

Thread P1 produced -6 into buf[93]

Thread P1 produced -86 into buf[94]

Thread P1 produced 94 into buf[95]

Thread P1 produced -14 into buf[96]

Thread P1 produced -32 into buf[97]

Thread P1 produced -91 into buf[98]

Thread P1 produced -24 into buf[99]

Thread P1 produced -28 into buf[100]

Thread P1 produced 44 into buf[101]

Thread P1 produced 38 into buf[102]

Thread P1 produced -15 into buf[103]

Thread P1 produced 10 into buf[104]

Thread P1 produced -92 into buf[105]

Thread P1 produced -62 into buf[106]

Thread P1 produced 73 into buf[107]

Thread P1 produced 42 into buf[108]

Thread P1 produced 19 into buf[109]

Thread P1 produced 93 into buf[110]

Thread P1 produced 5 into buf[111]

Thread P4 produced -76 into buf[112]

Thread P4 produced 28 into buf[113]

Thread P4 produced -10 into buf[114]

Thread P4 produced -24 into buf[115]

Thread P4 produced 88 into buf[116]

Thread P4 produced -67 into buf[117]

Thread P4 produced -75 into buf[118]

Thread P4 produced -94 into buf[119]

Thread P4 produced 54 into buf[120]

Thread P4 produced 39 into buf[121]

Thread P4 produced -11 into buf[122]

Thread P4 produced -78 into buf[123]

Thread P4 produced 68 into buf[124]

Thread P5 consumed 68 from buf[124]

Thread P5 accessed CR2 data (used)

Thread P2 consumed -78 from buf[123]

Thread P5 released semaphore for P2

Thread P6 accessed CR2 data (edited)

Thread P4 produced -24 into buf[123]

Thread P2 released semaphore for P5

Thread P2 got semaphore released from P5

Thread P4 produced 79 into buf[124]

Thread P5 got semaphore released from P2

Thread P4 produced 50 into buf[125]

Thread P4 produced 37 into buf[126]

Thread P4 produced -72 into buf[127]

Thread P4 produced -9 into buf[128]

Thread P4 produced 60 into buf[129]

Thread P4 produced -89 into buf[130]

Thread P4 produced 78 into buf[131]

Thread P4 produced 37 into buf[132]

Thread P4 produced 93 into buf[133]

Thread P1 produced -20 into buf[134]

Thread P1 produced 72 into buf[135]

Thread P1 produced 64 into buf[136]

Thread P1 produced 65 into buf[137]

Thread P1 produced -14 into buf[138]

Thread P1 produced -25 into buf[139]

Thread P1 produced -11 into buf[140]

Thread P1 produced -40 into buf[141]

Thread P1 produced -4 into buf[142]

Thread P1 produced 30 into buf[143]

Thread P1 produced 49 into buf[144]

Thread P1 produced -70 into buf[145]

Thread P4 produced 46 into buf[146]

Thread P4 produced 13 into buf[147]

Thread P4 produced 51 into buf[148]

Thread P4 produced 14 into buf[149]

Thread P4 produced -81 into buf[150]

Thread P4 produced 81 into buf[151]

Thread P4 produced 61 into buf[152]

Thread P4 produced -96 into buf[153]

Thread P4 produced 58 into buf[154]

Thread P4 produced -7 into buf[155]

Thread P4 produced -99 into buf[156]

Thread P4 produced 16 into buf[157]

Thread P4 produced 11 into buf[158]

Thread P4 produced -36 into buf[159]

Thread P4 produced -68 into buf[160]

Thread P4 produced 31 into buf[161]

Thread P4 produced 91 into buf[162]

Thread P4 produced -90 into buf[163]

Thread P4 produced -86 into buf[164]

Thread P4 produced -43 into buf[165]

Thread P4 produced 40 into buf[166]

Thread P4 produced -23 into buf[167]

Thread P4 produced 75 into buf[168]

Thread P4 produced 94 into buf[169]

Thread P4 produced 28 into buf[170]

Thread P4 produced 40 into buf[171]

Thread P4 produced 82 into buf[172]

Thread P4 produced 91 into buf[173]

Thread P4 produced 40 into buf[174]

Thread P4 produced -1 into buf[175]

Thread P4 produced 57 into buf[176]

Thread P4 produced -68 into buf[177]

Thread P4 produced 14 into buf[178]

Thread P4 produced -10 into buf[179]

Thread P4 produced -38 into buf[180]

Thread P4 produced 43 into buf[181]

Thread P4 produced -24 into buf[182]

Thread P4 produced 21 into buf[183]

Thread P4 produced 5 into buf[184]

Thread P4 produced 36 into buf[185]

Thread P4 produced -89 into buf[186]

Thread P4 produced 65 into buf[187]

Thread P4 produced -9 into buf[188]

Thread P4 produced 40 into buf[189]

Thread P4 produced 35 into buf[190]

Thread P4 produced 67 into buf[191]

Thread P4 produced -69 into buf[192]

Thread P4 produced -79 into buf[193]

Thread P4 produced 60 into buf[194]

Thread P4 produced -94 into buf[195]

Thread P4 produced -41 into buf[196]

Thread P4 produced -96 into buf[197]

Thread P4 produced 11 into buf[198]

Thread P4 produced 7 into buf[199]

Thread P5 consumed 7 from buf[199]

Thread P5 accessed CR2 data (used)

Thread P2 consumed 11 from buf[198]

Thread P5 released semaphore for P2

Thread P6 accessed CR2 data (edited)

Thread P4 produced 54 into buf[198]

Thread P2 released semaphore for P5

Thread P4 produced 68 into buf[199]

Thread P2 got semaphore released from P5

Thread P5 got semaphore released from P2

Thread P4 produced -75 into buf[200]

Thread P4 produced 36 into buf[201]

Thread P4 produced 69 into buf[202]

Thread P4 produced -38 into buf[203]

Thread P1 produced 22 into buf[204]

Thread P1 produced -49 into buf[205]

Thread P1 produced 25 into buf[206]

Thread P1 produced -25 into buf[207]

Thread P1 produced 95 into buf[208]

Thread P1 produced -8 into buf[209]

Thread P1 produced -2 into buf[210]

Thread P1 produced 50 into buf[211]

Thread P1 produced -94 into buf[212]

Thread P1 produced 48 into buf[213]

Thread P1 produced 5 into buf[214]

Thread P1 produced -59 into buf[215]

Thread P1 produced 61 into buf[216]

Thread P1 produced 34 into buf[217]

Thread P1 produced -78 into buf[218]

Thread P1 produced 41 into buf[219]

Thread P1 produced -43 into buf[220]

Thread P1 produced 12 into buf[221]

Thread P1 produced -23 into buf[222]

Thread P1 produced 98 into buf[223]

Thread P1 produced 13 into buf[224]

Thread P1 produced 10 into buf[225]

Thread P1 produced -49 into buf[226]

Thread P1 produced -77 into buf[227]

Thread P1 produced -35 into buf[228]

Thread P1 produced -63 into buf[229]

Thread P1 produced 61 into buf[230]

Thread P1 produced -35 into buf[231]

Thread P1 produced 14 into buf[232]

Thread P1 produced -100 into buf[233]

Thread P1 produced 7 into buf[234]

Thread P1 produced 9 into buf[235]

Thread P1 produced 41 into buf[236]

Thread P1 produced -25 into buf[237]

Thread P1 produced 99 into buf[238]

Thread P1 produced -96 into buf[239]

Thread P1 produced 55 into buf[240]

Thread P1 produced 38 into buf[241]

Thread P1 produced 37 into buf[242]

Thread P1 produced -78 into buf[243]

Thread P1 produced 11 into buf[244]

Thread P1 produced 16 into buf[245]

Thread P1 produced -81 into buf[246]

Thread P4 produced -37 into buf[247]

Thread P4 produced 60 into buf[248]

Thread P4 produced 38 into buf[249]

Thread P4 produced -18 into buf[250]

Thread P4 produced 7 into buf[251]

Thread P4 produced 7 into buf[252]

Thread P4 produced -6 into buf[253]

Thread P4 produced -72 into buf[254]

Thread P4 produced -90 into buf[255]

Thread P4 produced -55 into buf[256]

Thread P5 consumed -55 from buf[256]

Thread P5 accessed CR2 data (used)

Thread P5 released semaphore for P2

Thread P2 consumed -90 from buf[255]

Thread P4 produced 37 into buf[255]

Thread P6 accessed CR2 data (edited)

Thread P2 released semaphore for P5

Thread P2 got semaphore released from P5

Thread P4 produced -7 into buf[256]

Thread P5 got semaphore released from P2

Thread P4 produced 2 into buf[257]

Thread P4 produced -73 into buf[258]

Thread P4 produced 71 into buf[259]

Thread P4 produced 35 into buf[260]

Thread P4 produced -2 into buf[261]

Thread P4 produced 67 into buf[262]

Thread P4 produced -30 into buf[263]

Thread P4 produced 33 into buf[264]

Thread P4 produced -4 into buf[265]

Thread P1 produced -42 into buf[266]

Thread P1 produced 36 into buf[267]

Thread P1 produced 58 into buf[268]

Thread P1 produced -86 into buf[269]

Thread P1 produced -10 into buf[270]

Thread P1 produced -63 into buf[271]

Thread P1 produced 63 into buf[272]

Thread P1 produced -15 into buf[273]

Thread P1 produced 59 into buf[274]

Thread P1 produced -86 into buf[275]

Thread P1 produced -17 into buf[276]

Thread P1 produced 37 into buf[277]

Thread P1 produced 2 into buf[278]

Thread P1 produced 2 into buf[279]

Thread P1 produced 85 into buf[280]

Thread P1 produced -73 into buf[281]

Thread P1 produced -86 into buf[282]

Thread P1 produced 32 into buf[283]

Thread P1 produced 96 into buf[284]

Thread P1 produced -28 into buf[285]

Thread P1 produced 66 into buf[286]

Thread P1 produced 39 into buf[287]

Thread P1 produced 95 into buf[288]

Thread P1 produced -82 into buf[289]

Thread P1 produced 54 into buf[290]

Thread P1 produced 64 into buf[291]

Thread P1 produced 55 into buf[292]

Thread P1 produced -72 into buf[293]

Thread P1 produced 4 into buf[294]

Thread P1 produced 9 into buf[295]

Thread P1 produced 2 into buf[296]

Thread P1 produced 7 into buf[297]

Thread P1 produced 7 into buf[298]

Thread P1 produced 39 into buf[299]

Thread P1 produced -34 into buf[300]

Producer P1 terminated the programm !

byteVariable: 69

shortVariable: 13907

intVariable: 1407205211

longVariable: 1559508127558454579

floatVariable: 0.6615651

doubleVariable: 0.6349044743664608

booleanVariable: false

charVariable: n