TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÁO CÁO CUỐI KỲ ALCHEMI

Giảng viên hướng dẫn: Đoàn Duy Trung

Nhóm SV thực hiện: Nhóm 9

Họ tên sinh viên	MSSV
1. Hoàng Phi Long	20185375
2. Nguyễn Hải Đăng	20185333
3. Đỗ Quang Hùng	20185365
4. Nguyễn Kim Long	20185379

Hà Nội, 5/2021

Phân công công việc

Phần việc	Thành viên thực hiện
Bài tập chung số 1	Nguyễn Kim Long
Bài tập chung số 2	Nguyễn Hải Đăng
Bài tập nhóm	Đỗ Quang Hùng Hoàng Phi Long
Bài tập thêm	Hoàng Phi Long
Tổng hợp, viết báo cáo	Đỗ Quang Hùng

I. BÀI THỰC HÀNH CHUNG

Bài 1: Liệt kê các số chính phương từ 1 đến n, với n nhập từ bàn phím. Phân chia đoạn [1, n] thành các đoạn để chạy trên các executor trong Lưới.

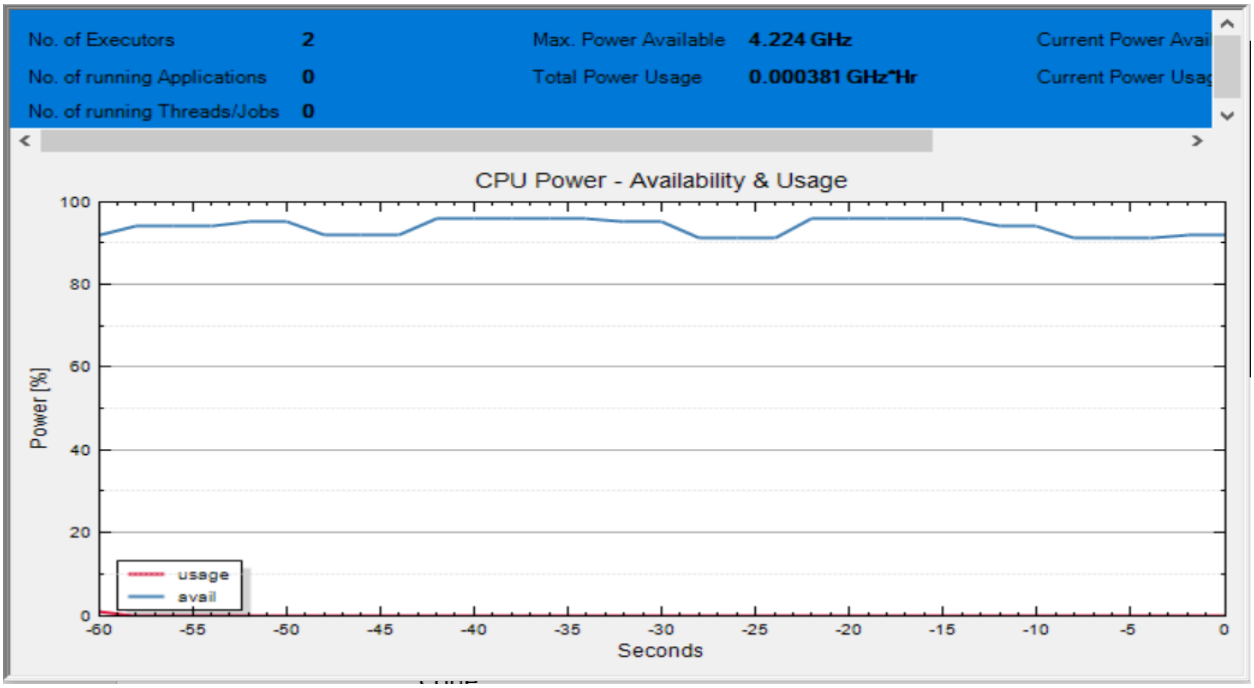
- **Code:**

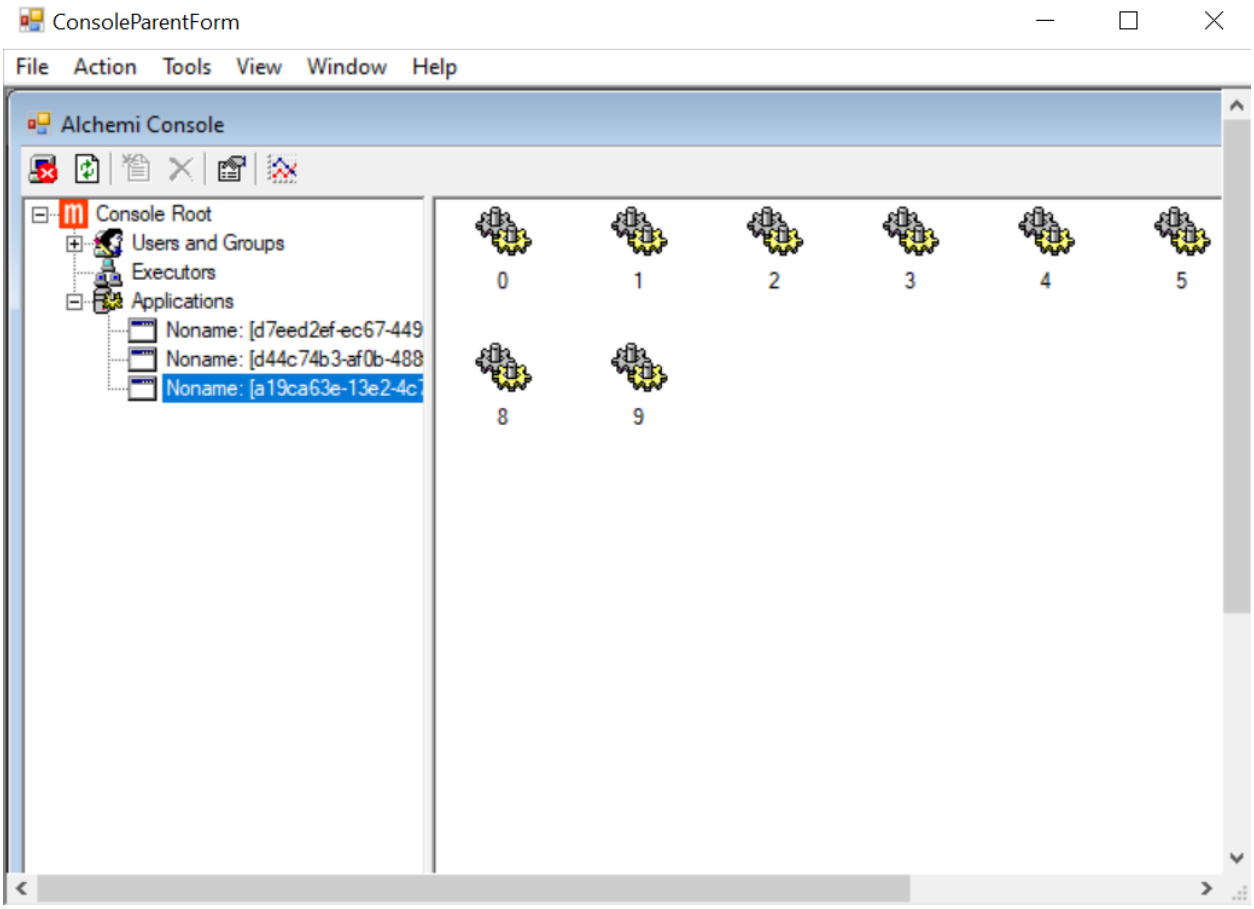
```
using System;
using System.Collections.Generic;
using System.Text;
using Alchemi.Core;
using Alchemi.Core.Owner;

namespace PrimeNumber
{

    class PrimeNumber : GApplication
    {
        public static GApplication App = new GApplication();
        private static DateTime start;

        [STAThread]
        static void Main(string[] args)
        {
            Console.WriteLine("[Integral Computation Grid Application]\n----------------
---------------\n");

            int n;
            Console.Write("Hay nhap n: ");
            n = Int32.Parse(Console.ReadLine());
            int m = (int)Math.Ceiling(Math.Sqrt(n));
            Console.Write("Hay nhap so luong: ");
            int Num_Of_Thread = Int32.Parse(Console.ReadLine());
            int NumThread;
            if (m % Num_Of_Thread != 0)
            {
                NumThread = m / Num_Of_Thread + 1;
            }
            else
                NumThread = m / Num_Of_Thread;

            SubWork[] Manager = new SubWork[Num_Of_Thread];
            for (int j = 0; j < Num_Of_Thread; j++)
            {
                if (j == (Num_Of_Thread - 1))
                {
                    App.Threads.Add(new SubWork(j, NumThread * j, m, n));
                }
                else
                {
                    App.Threads.Add(new SubWork(j, NumThread * j, NumThread * (j + 1) -
1, n));
```

```
                }
            }
            App.Connection = new GConnection("localhost", 9000, "user", "user");
            App.Manifest.Add(new ModuleDependency(typeof(SubWork).Module));
            App.ThreadFinish += new GThreadFinish(App_ThreadFinish);
            App.ApplicationFinish += new GApplicationFinish(App_ApplicationFinish);
            start = DateTime.Now;
            Console.WriteLine("Thread started!");
            App.Start();
            Console.ReadLine();
            App.Stop();

        }

        private static void App_ThreadFinish(GThread thread)
        {
            SubWork sw = (SubWork)thread;
            for (int i = sw.start_num; i <= sw.end_num; i++)
            {
                if (i != 0)
                {
                    if (i * i <= sw.n)
                        Console.Write(i*i + " ");
                }
            }
            Console.WriteLine(" \nThread so {0} da hoan thanh! ", sw.ThreadId);
        }

        private static void App_ApplicationFinish()
        {

            Console.WriteLine("Hoan thanh sau {0} seconds.", DateTime.Now - start);
        }
    }
    [Serializable]
    class SubWork : GThread
    {
        public int ThreadId, n, start_num, end_num;
        public SubWork(int ThreadId, int start_num, int end_num, int n)
        {
            this.ThreadId = ThreadId;
            this.start_num = start_num;
            this.end_num = end_num;
            this.n = n;
        }

        public override void Start()
        {
            Console.WriteLine(" \nThread {0} vua lam xong viec!!!", ThreadId);
        }
    }
}
```

- **Màn hình chạy và số luồng**

```
[Integral Computation Grid Application]
--------------------------------

Hay nhap n: 1000
Hay nhap so luong: 10
Thread started!
1 4 9
Thread so 0 da hoan thanh!
16 25 36 49
Thread so 1 da hoan thanh!
64 81 100 121
Thread so 2 da hoan thanh!
144 169 196 225
Thread so 3 da hoan thanh!
256 289 324 361
Thread so 4 da hoan thanh!
400 441 484 529
Thread so 5 da hoan thanh!
576 625 676 729
Thread so 6 da hoan thanh!
784 841 900 961
Thread so 7 da hoan thanh!

Thread so 8 da hoan thanh!

Thread so 9 da hoan thanh!
Hoan thanh sau 00:00:02.8689914 seconds.
```




ConsoleParentForm
File
Action
Tools
View
Window
Help
Alchemi Console
Console Root
Users and Groups
Executors
Applications
Noname: [d7eed2ef-ec67-449
Noname: [d44c74b3-af0b-488
Noname: [a19ca63e-13e2-4c
0
1
2
3
4
5
8
9

Bài 2: Cho n nhập từ bàn phím. Tính gần đúng tích phân sau:

$$\int_0^n f(x)dx$$

Ở đó hàm f(x) tùy ý
Phân chia [0, n] vào các Executor để chạy trong lưới.

Phần này, nhóm em sử dụng công thức hình thang trong sách Giải tích số thầy Lê Trọng Vinh để tính tích phân của hàm số $3x^2 + 2x + 1$. Nhóm em chọn hàm số này để thuận tiện cho việc tính toán và kiểm soát sai số

- **Code**

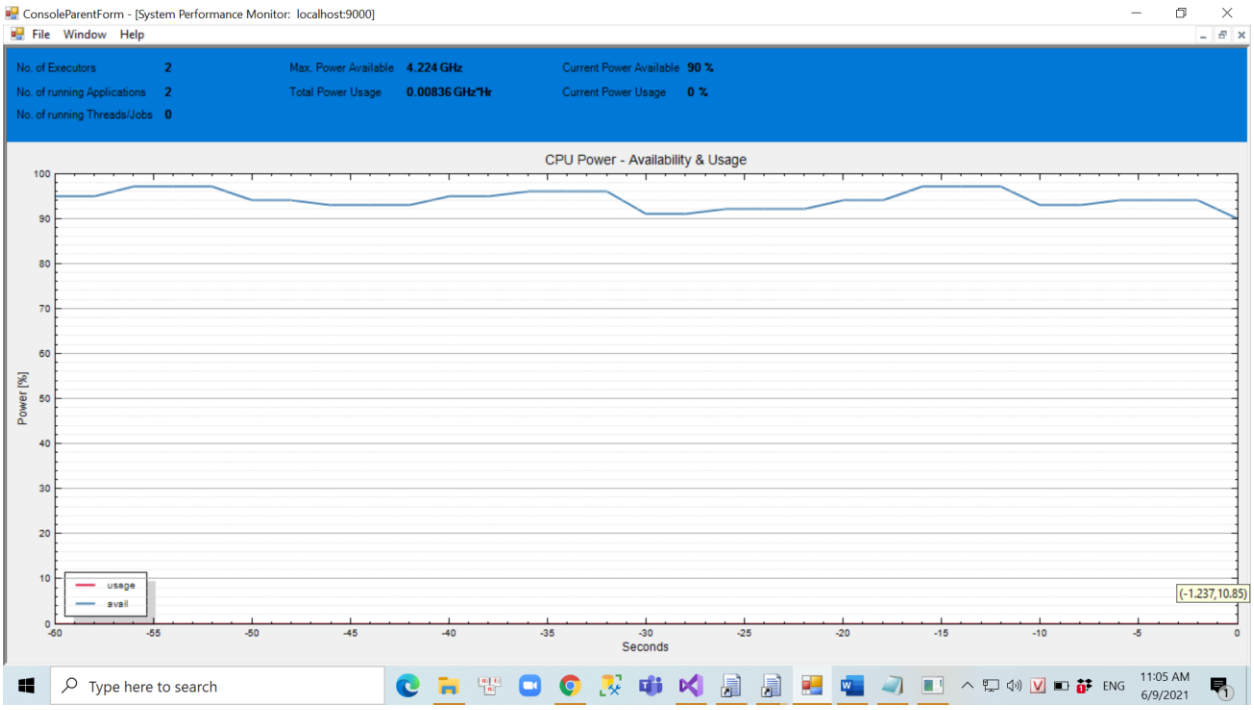
```
using System;
using System.Collections.Generic;
using System.Text;
using Alchemi.Core;
using Alchemi.Core.Owner;

namespace PrimeNumber
{
    public static class Globals
    {
        public static double big_sum = 0;
    }

    class PrimeNumber : GApplication
    {
        public static GApplication App = new GApplication();
        private static DateTime start;

        [STAThread]
        static void Main(string[] args)
        {
            Console.WriteLine("[Integral Computation Grid Application]\n-----------------
---------------\n");
            int n;
            Console.Write("Hay nhap n: ");
            n = Int32.Parse(Console.ReadLine());
            Console.Write("Hay nhap do chinh xac: ");
            int m = Int32.Parse(Console.ReadLine());
            int p = (int)Math.Ceiling(Math.Sqrt(Math.Pow(10, m) * Math.Pow(n, 3) / 2));
            Console.Write("Hay nhap so luong phep tinh cua moi thread: ");
            int NumThread = Int32.Parse(Console.ReadLine());
            int Num_Of_Thread = (p / NumThread) + 1;
            for (int j = 0; j < Num_Of_Thread; j++)
            {
                if (j == (Num_Of_Thread - 1))
                {
                    App.Threads.Add(new SubWork(j, NumThread * j, p, p, n));
                }
                else
                {
                    App.Threads.Add(new SubWork(j, NumThread * j, NumThread * (j + 1) -
1, p, n));
```

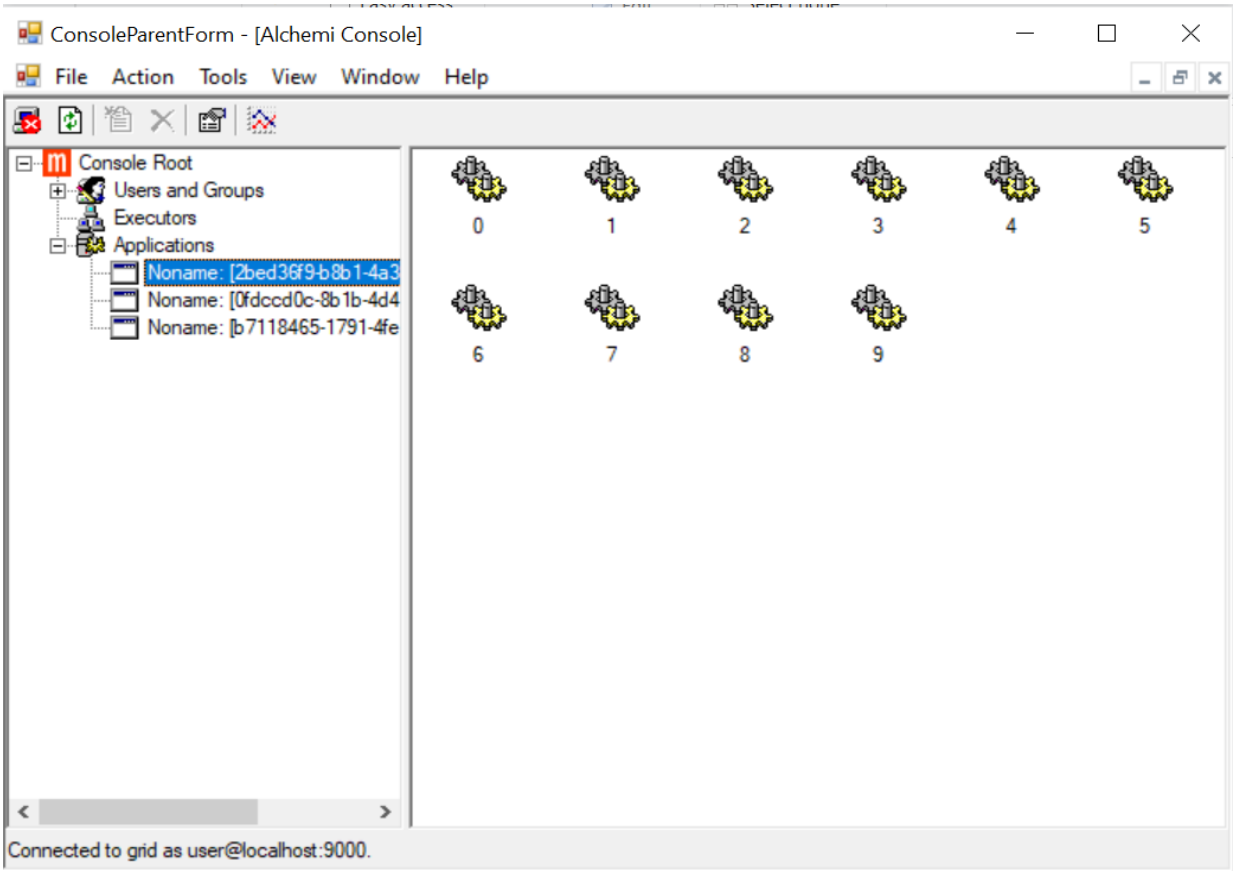
```
                }
            }
            App.Connection = new GConnection("localhost", 9000, "user", "user");
            App.Manifest.Add(new ModuleDependency(typeof(SubWork).Module));
            App.ThreadFinish += new GThreadFinish(App_ThreadFinish);
            App.ApplicationFinish += new GApplicationFinish(App_ApplicationFinish);
            start = DateTime.Now;
            Console.WriteLine("Thread started!");
            App.Start();
            Console.ReadLine();

            App.Stop();
        }

        private static void App_ThreadFinish(GThread thread)
        {
            SubWork sw = (SubWork)thread;
            Globals.big_sum += sw.Sub_Sum;
            Console.WriteLine("Thread so {0} da hoan thanh!", sw.ThreadId);
            Console.WriteLine("Gia tri cua tich phan: " + Globals.big_sum);
        }

        private static void App_ApplicationFinish()
        {
            Console.WriteLine("Hoan thanh sau {0} seconds.", DateTime.Now - start);
        }
    }
    [Serializable]
    class SubWork : GThread
    {
        public int ThreadId, p, n, start_num, end_num;
        public double Sub_Sum = 0;
        public SubWork(int ThreadId, int start_num, int end_num, int p, int n)
        {
            this.ThreadId = ThreadId;
            this.start_num = start_num;
            this.end_num = end_num;
            this.p = p;
            this.n = n;
        }
        static double Function(double x)
        {
            return 3 * Math.Pow(x, 2) + 2 * x + 1;
        }

        public override void Start()
        {
            for (int i = start_num; i <= end_num; i++)
            {
                double x = i * (n * 1.0 / p);
                if (i == 0 || i == p) Sub_Sum += Function(x);
                else Sub_Sum += 2 * Function(x);
            }
            Sub_Sum = n * Sub_Sum / (2 * p);
        }
    }
}
```

Màn hình chạy và số luồng

```
[Integral Computation Grid Application]
-------------------------------

Hay nhap n: 10
Hay nhap do chinh xac: 3
Hay nhap so luong phep tinh cua moi thread: 3
Thread started!
Thread so 0 da hoan thanh!
Gia tri cua tich phan: 0.0365499745457044
Thread so 1 da hoan thanh!
Gia tri cua tich phan: 0.0841334124282965
Thread so 2 da hoan thanh!
Gia tri cua tich phan: 0.13614463990963
Thread so 3 da hoan thanh!
Gia tri cua tich phan: 0.193040130144216
Thread so 4 da hoan thanh!
Gia tri cua tich phan: 0.255276356286562
Thread so 5 da hoan thanh!
Gia tri cua tich phan: 0.323309791491178
Thread so 6 da hoan thanh!
Gia tri cua tich phan: 0.397596908912574
Thread so 7 da hoan thanh!
Gia tri cua tich phan: 0.478594181705259
Thread so 8 da hoan thanh!
Gia tri cua tich phan: 0.566758083023743
Thread so 9 da hoan thanh!
Gia tri cua tich phan: 0.662545086022535
Thread so 10 da hoan thanh!
Gia tri cua tich phan: 0.766411663856145
Thread so 11 da hoan thanh!
Gia tri cua tich phan: 0.878814289679081
Thread so 12 da hoan thanh!
Gia tri cua tich phan: 1.00020943664585
Thread so 13 da hoan thanh!
Gia tri cua tich phan: 1.13105357791097
Thread so 14 da hoan thanh!
Gia tri cua tich phan: 1.27180318662895
Thread so 15 da hoan thanh!
```

```
Thread so 220 da hoan thanh!
Gia tri cua tich phan: 916.244873637411
Thread so 221 da hoan thanh!
Gia tri cua tich phan: 928.26300907915
Thread so 222 da hoan thanh!
Gia tri cua tich phan: 940.385996404479
Thread so 223 da hoan thanh!
Gia tri cua tich phan: 952.614292086554
Thread so 224 da hoan thanh!
Gia tri cua tich phan: 964.948352598529
Thread so 225 da hoan thanh!
Gia tri cua tich phan: 977.388634413558
Thread so 226 da hoan thanh!
Gia tri cua tich phan: 989.935594004796
Thread so 227 da hoan thanh!
Gia tri cua tich phan: 1002.5896878454
Thread so 228 da hoan thanh!
Gia tri cua tich phan: 1015.35137240852
Thread so 229 da hoan thanh!
Gia tri cua tich phan: 1028.22110416731
Thread so 230 da hoan thanh!
Gia tri cua tich phan: 1041.19933959493
Thread so 231 da hoan thanh!
Gia tri cua tich phan: 1054.28653516453
Thread so 232 da hoan thanh!
Gia tri cua tich phan: 1067.48314734926
Thread so 233 da hoan thanh!
Gia tri cua tich phan: 1080.78963262229
Thread so 234 da hoan thanh!
Gia tri cua tich phan: 1094.20644745676
Thread so 235 da hoan thanh!
Gia tri cua tich phan: 1107.73404832583
Thread so 236 da hoan thanh!
Gia tri cua tich phan: 1110.00099747837
Hoan thanh sau 00:00:04.7113598 seconds.
```


ConsoleParentForm - [System Performance Monitor: localhost:9000]
File Window Help
No. of Executors 2
No. of running Applications 2
No. of running Threads/Jobs 0
Max. Power Available 4.224 GHz
Total Power Usage 0.00836 GHz*Hr
Current Power Available 90 %
Current Power Usage 0 %
CPU Power - Availability & Usage
Power [%]
Seconds
usage
avail
(-1.237,10.85)
Type here to search
ENG
11:05 AM
6/9/2021


ConsoleParentForm - [Alchemi Console]
File Action Tools View Window Help
Console Root
Users and Groups
Executors
Applications
Noname: [2bed36f9-b8b1-4a3
Noname: [0fdccd0c-8b1b-4d4
Noname: [b7118465-1791-4fe
0
1
2
3
4
5
6
7
8
9
Connected to grid as user@localhost:9000.

II. Bài tập theo nhóm

Đề 1:

Đọc 1 văn bản dưới dạng file input.txt, thực hiện hàm số f(x) = 2x + 1, ở đó x là ký tự mã ASCII của từng ký tự trong file input.txt và in ký tự mã ASCII tương ứng f(x) ra file output.txt. Yêu cầu phân chia file input.txt ra thành k luồng (k nhập từ bàn phím), theo kích thước của tệp tin input.txt để thực hiện trên các nút trong lưới tính toán

- **Code**

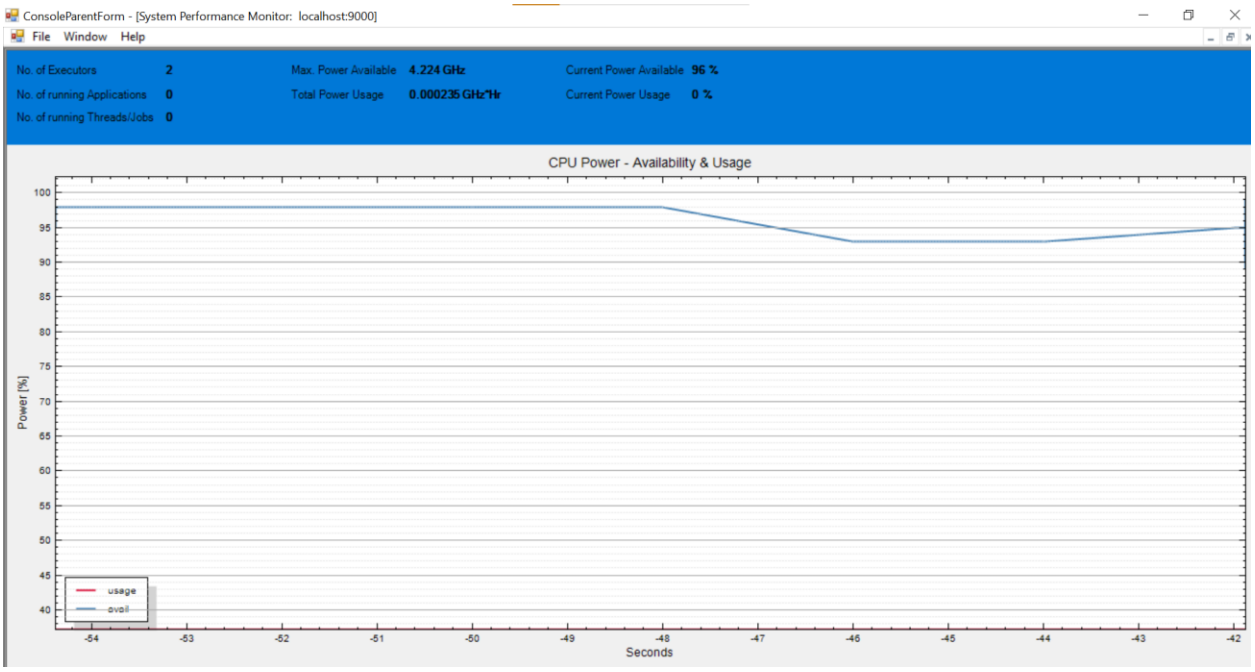
```
using System;
using System.IO;
using System.Text;
using Alchemi.Core;
using Alchemi.Core.Owner;

namespace HelloWorld
{
    class Program
    {
        [Serializable]
        class Encode : GThread
        {
            public readonly int[] Candidate = new int[BytesPerThread];
            public int STT = -1;

            public Encode(int[] candidate, int stt)
            {
                for (int i = 0; i < candidate.Length; i++)
                    Candidate[i] = candidate[i];
                STT = stt;

            }
            public override void Start()
            {
                for (int i = 0; i < Candidate.Length; i++)
                {
                    if (Candidate[i] == 255)
                        break;
                    else
                        Candidate[i] = (Candidate[i] * 2 + 1) % 127;
                }
            }
        }
        public static class Globals
        {
            public static byte[] A;
        }

        static int NumThread = 10;
        static int BytesPerThread;
        static int dem = 0;
        class Generator
        {
            public static GApplication App = new GApplication();
```

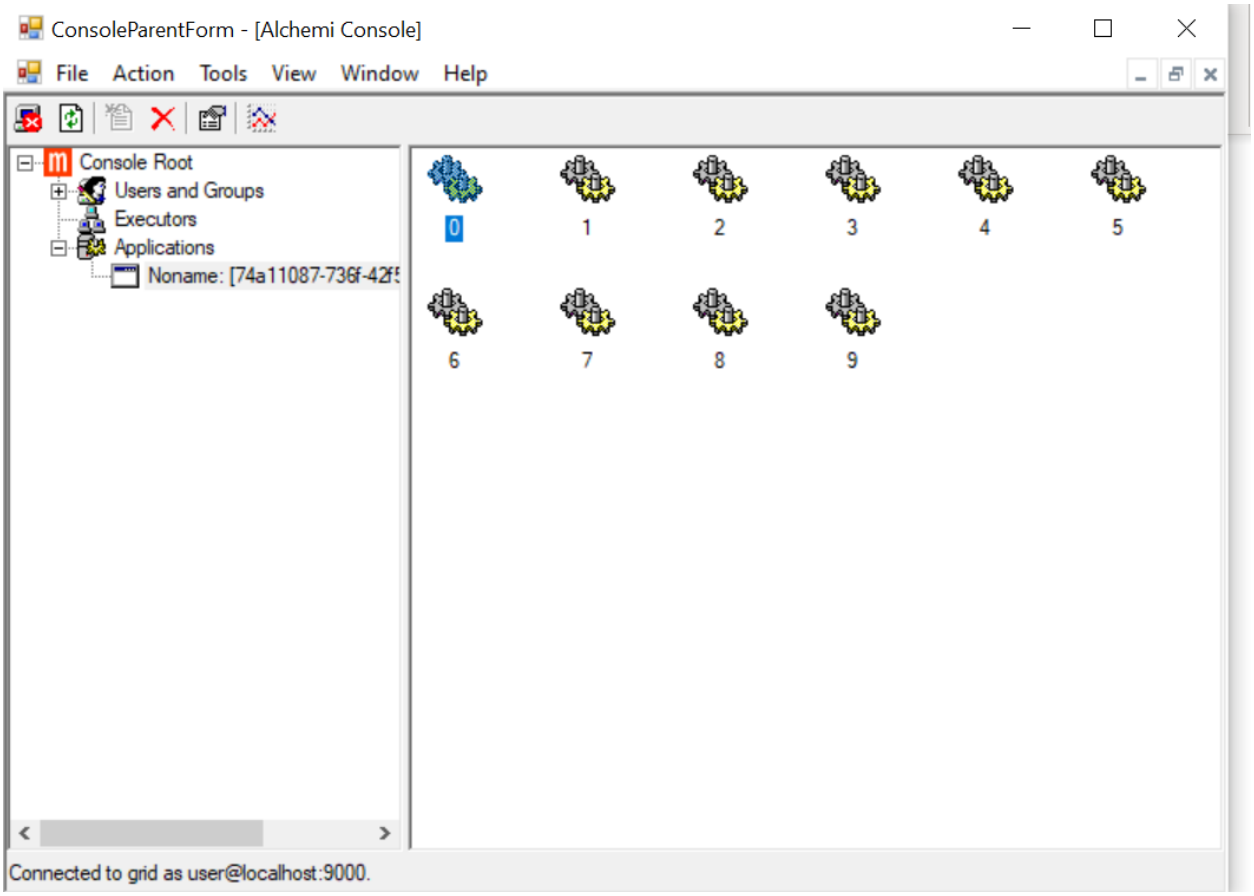
```
            private static DateTime start;

            [STAThread]
            static void Main(string[] args)
            {
                Console.WriteLine("[Integral Computation Grid Application]\n------------
-------------------\n");
                Console.Write("Nhap so luong: ");
                NumThread = Convert.ToInt32(Console.ReadLine());
                Globals.A = File.ReadAllBytes("D:\\Hoc
tap\\20202\\TTSS\\testalchemi.txt");
                BytesPerThread = (Globals.A.Length / (NumThread)) + 1;
                Console.WriteLine("\nKich thuoc cua file la {0} bytes",
Globals.A.Length);
                Console.WriteLine("Chia moi luong {0} bytes", BytesPerThread);
                Console.WriteLine("\n---------- Start ----------\n");

                if (BytesPerThread > NumThread)
                {
                    int[] B = new int[BytesPerThread];

                    for (int i = 0; i < NumThread - 1; i++)
                    {
                        for (int j = 0; j < BytesPerThread; j++)
                        {
                            B[j] = Convert.ToInt32(Globals.A[i * BytesPerThread + j]);

                        }
                        App.Threads.Add(new Encode(B, i));
                    }

                    for (int j = 0; j < (BytesPerThread); j++)
                    {
                        B[j] = 255;
                    }

                    try
                    {
                        for (int j = 0; j < BytesPerThread - 1; j++)
                        {

                            B[j] = Convert.ToInt32(Globals.A[(NumThread - 1) *
BytesPerThread + j]);
                        }
                    }
                    catch
                    {

                    }

                    App.Threads.Add(new Encode(B, (NumThread - 1)));
                    App.Connection = new GConnection("localhost", 9000, "user", "user");
                    App.Manifest.Add(new ModuleDependency(typeof(Encode).Module));
                    App.ThreadFinish += new GThreadFinish(App_ThreadFinish);
                    App.ApplicationFinish += new
GApplicationFinish(App_ApplicationFinish);
                    start = DateTime.Now;
                    App.Start();
```

```
                    while (dem < NumThread)
                    {

                    }

                    App.Stop();
                    Console.WriteLine("\n---------- Finish ----------");
                    Console.ReadLine();

                    char[] c = new char[Globals.A.Length];

                    for (int i = 0; i < c.Length; i++)
                    {
                        c[i] = Convert.ToChar(Globals.A[i]);

                    }

                    FileStream sb = new FileStream("D:\\Hoc
tap\\20202\\TTSS\\output.txt", FileMode.OpenOrCreate);

                    StreamWriter sw = new StreamWriter(sb);
                    sw.Write(c, 0, c.Length);
                    sw.Close();
                }
                else
                    Console.WriteLine("Chuong trinh khong the chay vi BytesPerThread <
NumThread ({0} < {1})",
                            BytesPerThread, NumThread);
            }

            private static void App_ApplicationFinish()
            {
                Console.WriteLine("Hoan thanh sau {0} seconds.", DateTime.Now - start);
            }

            private static void App_ThreadFinish(GThread thread)
            {
                Encode enc = (Encode)thread;
                dem += 1;
                Console.WriteLine("Da xu ly xong {0} luong", dem);
                if(enc.STT == NumThread - 1)
                for (int i = 0; i < Globals.A.Length - (NumThread - 1) * BytesPerThread;
i++)
                {
                        Globals.A[enc.STT * (BytesPerThread) + i] =
(byte)enc.Candidate[i];
                }
                else
                for (int i = 0; i < enc.Candidate.Length; i++)
                {
                        Globals.A[enc.STT * (BytesPerThread) + i] =
(byte)enc.Candidate[i];
                }
            }
        }
    }
}
```

- **Màn hình chạy và số luồng**

```
[Integral Computation Grid Application]
--------------------------------

Nhap so luong luong: 10

Kich thuoc cua file la 4640 bytes
Chia moi luong 465 bytes

---------- Start ----------

Da xu ly xong 1 luong
Da xu ly xong 2 luong
Da xu ly xong 3 luong
Da xu ly xong 4 luong
Da xu ly xong 5 luong
Da xu ly xong 6 luong
Da xu ly xong 7 luong
Da xu ly xong 8 luong
Da xu ly xong 9 luong
Da xu ly xong 10 luong

---------- Finish ----------
Hoan thanh sau 00:00:02.9482814 seconds.
```




ConsoleParentForm - [Alchemi Console]
File Action Tools View Window Help
Console Root
Users and Groups
Executors
Applications
Noname: [74a11087-736f-42f5
0 1 2 3 4 5
6 7 8 9
Connected to grid as user@localhost:9000.

III. Bài tập thêm

Tính căn bậc hai của số nguyên k bất kỳ chính xác đến n chữ số sau phần thập phân (n – nhập từ bàn phím)

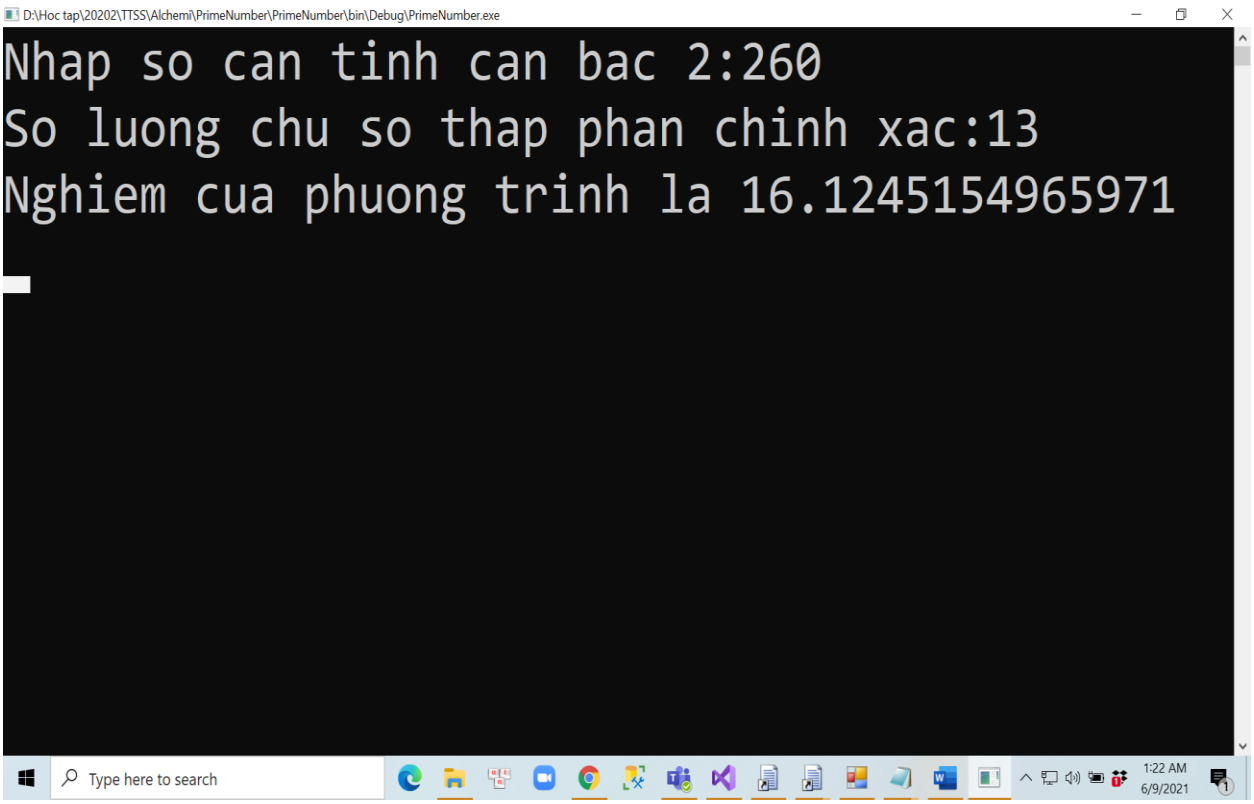
- **Code**

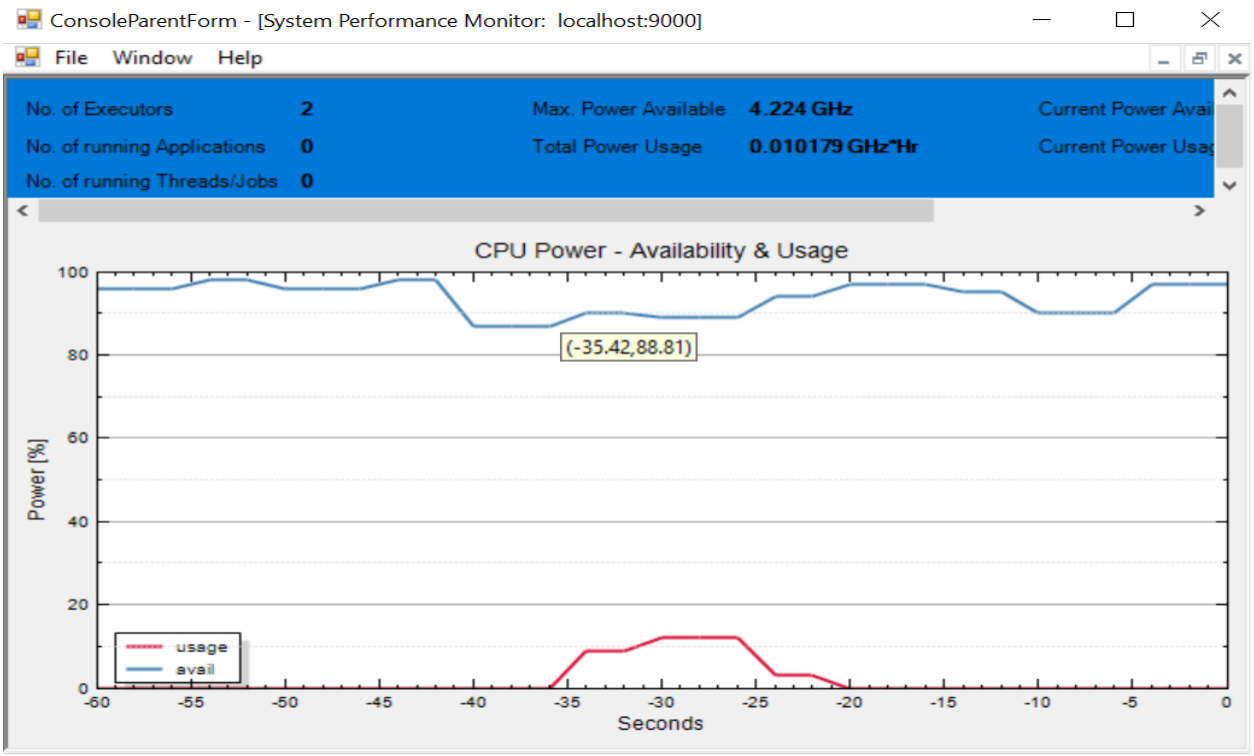
```
using System;
using System.Collections.Generic;
using System.Text;
using Alchemi.Core;
using Alchemi.Core.Owner;
using System.Threading;
namespace Chiadoi
{
    [Serializable]
    class kiemtra : GThread
    {
        public readonly double start, end, k;
        public int ketqua;

        public kiemtra(double x, double y, double z)
        {
            start = x;
            end = y;
            k = z;
        }
        public override void Start()
        {
            //hàm sử dụng để kiểm tra(x, y) có phải là khoảng phân ly nghiệm của phương
trình x* x-n = 0 ko?
            //return 0 tức (x, y) ko phải khoảng phân ly nghiệm
            //return 2 tức x là nghiệm của phương trình
            //return 1 tức (x, y) là khoảng phân ly nghiệm
            //return 3 tức y là nghiệm của phương trình
            if (start * start - k == 0)
                ketqua = 2;
            if (end * end - k == 0)
                ketqua = 3;
            ketqua = Convert.ToInt32(start * start - k < 0 && end * end - k > 0);
        }

        class ChiaDoiGenerator
        {
            public static GApplication App = new GApplication();
            public static double saiso, a, b;
            public static bool dem = false, ketthuc = false;
            [STAThread]
            static void Main(string[] args)
            {
                int n, k;
                int NumThread = 10;

                Console.Write("Nhap so can tinh can bac 2:");
                k = Int32.Parse(Console.ReadLine());
                Console.Write("So luong chu so thap phan chinh xac:");
```

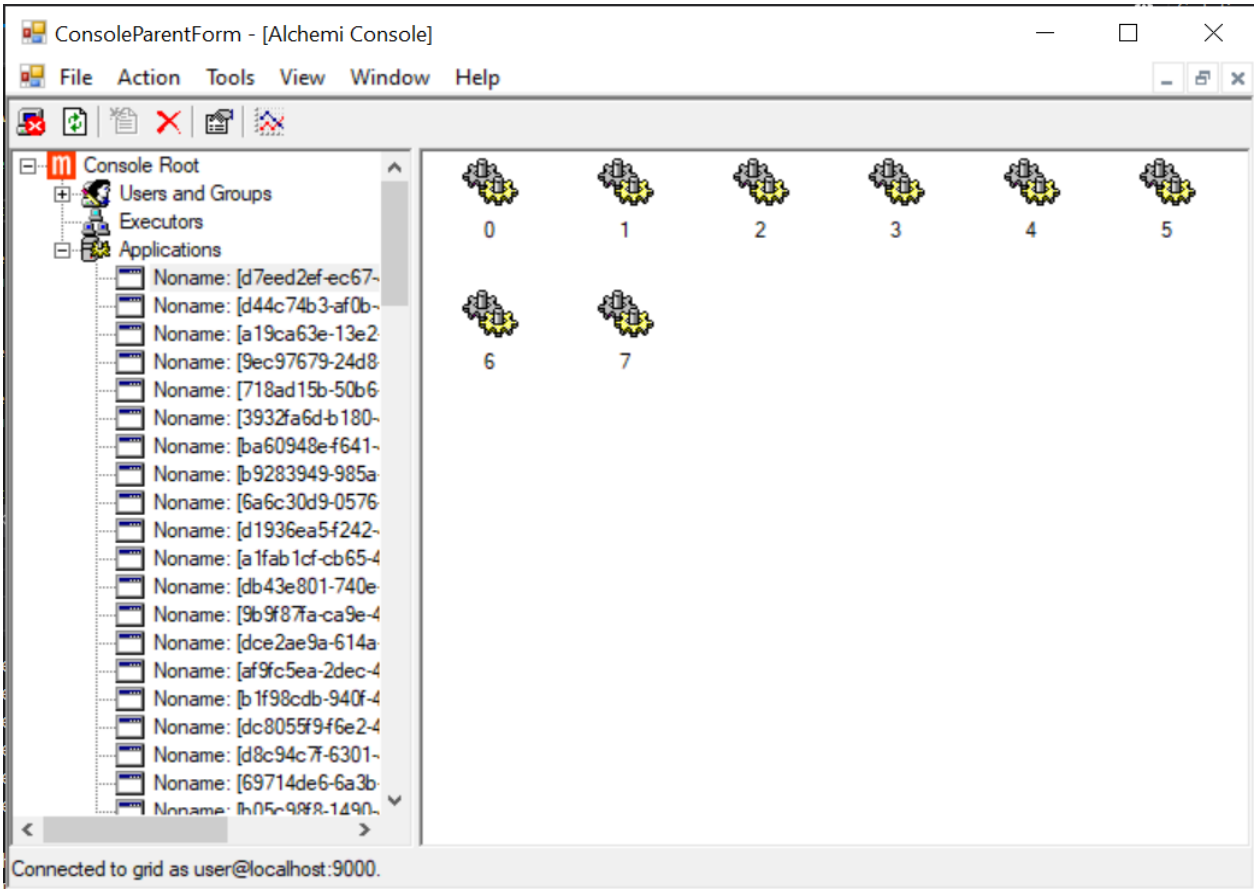
```
        n = Int32.Parse(Console.ReadLine());
        saiso = 1.0 / Math.Pow(10, n);
        a = 0;
        b = k;

        double step, start, end;
        while (b - a > saiso && ketthuc == false)
        {
            App = new GApplication();
            Init();
            dem = false;
            step = (b - a) / NumThread;
            end = a;
            for (int i = 0; i < NumThread; i++)
            {
                start = end;
                end = start + step;
                App.Threads.Add(new kiemtra(start, end, k));
            }
            App.Start();
            while (dem == false)
            { }

            dem = false;
            App.Stop();

        }
        if (ketthuc == false)
            Console.WriteLine("Nghiem cua phuong trinh la {0}", (a + b) / 2);
            Console.ReadKey();
    }
    private static void Init()
    {
        // specify connection properties
        App.Connection = new GConnection("localhost", 9000,
        "user", "user");
        // grid thread needs to
        App.Manifest.Add(new
        ModuleDependency(typeof(kiemtra).Module));
        // subscribe to ThreadFinish event
        App.ThreadFinish += new GThreadFinish(App_ThreadFinish);
    }
    private static void App_ThreadFinish(GThread thread)
    {
        kiemtra kt = (kiemtra)thread;

        if (kt.start * kt.start - kt.k == 0)
        {
            Console.WriteLine("Nghiem cua phuong trinh la {0}", kt.start);
            ketthuc = true;
            dem = true;
        }
        if (kt.end * kt.end - kt.k == 0)
        {
            Console.WriteLine("Nghiem cua phuong trinh la {0}", kt.end);
            ketthuc = true;
            dem = true;
        }
```

```
                if (kt.start * kt.start - kt.k < 0 && kt.end * kt.end - kt.k > 0)
                {
                    a = kt.start;
                    b = kt.end;
                    dem = true;
                }
            }
        }
    }
}
```

- **Màn hình chạy và số luồng**




ConsoleParentForm - [System Performance Monitor: localhost:9000]
File Window Help
No. of Executors 2
No. of running Applications 0
No. of running Threads/Jobs 0
Max. Power Available 4.224 GHz
Total Power Usage 0.010179 GHz*Hr
Current Power Avai
Current Power Usag
CPU Power - Availability & Usage
(-35.42,88.81)
Power [%]
usage
avail
Seconds


ConsoleParentForm - [Alchemi Console]
File Action Tools View Window Help
Console Root
Users and Groups
Executors
Applications
Noname: [d7eed2ef-ec67-
Noname: [d44c74b3-af0b-
Noname: [a19ca63e-13e2
Noname: [9ec97679-24d8
Noname: [718ad15b-50b6
Noname: [3932fa6d-b180-
Noname: [ba60948e-f641-
Noname: [b9283949-985a
Noname: [6a6c30d9-0576
Noname: [d1936ea5-f242-
Noname: [a1fab1cf-cb65-4
Noname: [db43e801-740e
Noname: [9b9f87fa-ca9e-4
Noname: [dce2ae9a-614a
Noname: [af9fc5ea-2dec-4
Noname: [b1f98cdb-940f-4
Noname: [dc8055f9-f6e2-4
Noname: [d8c94c7f-6301-
Noname: [69714de6-6a3b
0 1 2 3 4 5 6 7
Connected to grid as user@localhost:9000.