

به نام خدا

پژوهش: نمایش اعداد اول از بازه عددی ۱ تا ۵۰۰

گروه: یزدان قربانی و مبین جلیلی

استاد: الهام تشخیصی

وظیفه / برنامه نویس	یزدان قربانی	مبین جلیلی
برنامه نویسی پروژه اصلی	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
نوشتن برنامه با روش های دیگر	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
نوشتن الگوریتم و طراحی فلوچارت	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
نوشتن داکيومنت	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

روش اول

```
0 references
internal class PrimeNumberChecker
{
    0 references
    static void Main(string[] args)
    {
        List<int> res = new List<int>();
        for (int number = 1; number <= 500; number++)
        {
            for (int checker = 1; checker <= number; checker++)
            {
                if (number != 1)
                {
                    if (number % checker == 0)
                    {
                        res.Add(checker);
                    }
                }
                else
                {
                    goto end_of_outter_loop;
                }
            }
            if (res.Count > 2)
            {
                Console.Write("{0} is not prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
            else
            {
                Console.Write("{0} is prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
        }
        end_of_outter_loop: { }
    }
    Console.ReadKey();
}
```

توضیح خط به خط

بعد از تعریف کلاس اینترنال PrimeNumberChecker و تابع Main برای اجرای کد یک لیست با دیتا تایپ درونی int تعریف کرده با نام res. سپس یک for loop طراحی کرده با `number = 1` و دیتا تایپ int را برای بهینه سازی تعریف می کنیم سپس شرط `number <= 500` را گذاشته و تایین می کنیم `number++` تا نامبر بعد از هر گشت یکی اضافه شود

همین طور یک for loop دیگر را تعریف کرده و می گوئیم `checker = 1` با دیتا تایپ int شرط می کنیم `checker <= number` و برای پایان دادن به حلقه می گوئیم `checker++`

یک if برای تشخیص عدد یک و غیر یک تایین کرده تا ریسپانسیو بودن پروژه ثبات بهتری یابد

می گوئیم اگر number یک نبود وارد شو و اگر تقسیم number بر checker باقی مانده صفر داشت checker را به عنوان یکی از شمارنده های عدد number به لیست res اضافه کن

حال اگر number برابر با یک بود با کی ورد goto به تابع `end_of_outter_loop` که در پایان حلقه بیرون تعریف شده می رود تا چرخه با عدد بعدی تکرار شود

در صورتی که باقی مانده تقسیم number بر checker غیر صفر بود به چرخه درونی بازگشته تا حلقه درونی با عددی دیگر تکرار شود

در پایان حلقه درونی به `if(res.Count>2)` می رسیم که میگوید اگر تعداد شمارنده های عدد مورد نظر number بیشتر از ۲ بود این عدد اول نیست و در غیر این صورت می گوید عدد مورد نظر number عددی اول است و در پایان هر شرط لیست res به طور کامل پاک سازی شده تا در ادامه قابل استفاده و مفید باشد وگرنه نمی تواند در شرط های پایانی جواب درستی بدهد

و در پایان حلقه بیرونی از `Console.ReadKey();` استفاده شده تا برنامه منتظر واکنش کاربر بماند و بعد اجازه خروج بخواند

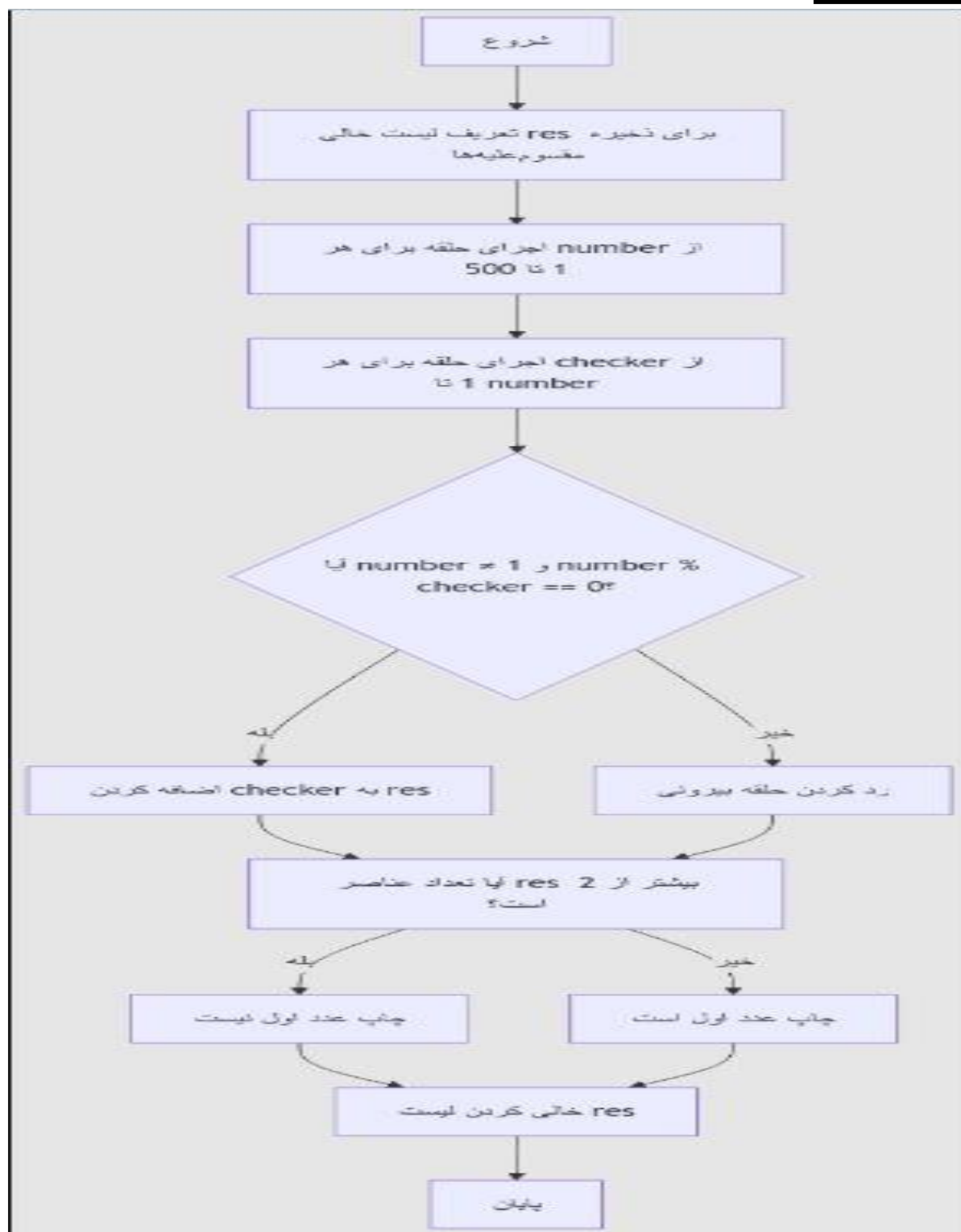
خروجی برنامه

```
E:\Yazdan_projects\cs_projects\findPrimeNumbers1\findPrimeNumbers1\bin\Debug\net8.0\findPrimeNumbers1.exe
449 is prime
450 is not prime
451 is not prime
452 is not prime
453 is not prime
454 is not prime
455 is not prime
456 is not prime
457 is prime
458 is not prime
459 is not prime
460 is not prime
461 is prime
462 is not prime
463 is prime
464 is not prime
465 is not prime
466 is not prime
467 is prime
468 is not prime
469 is not prime
470 is not prime
471 is not prime
472 is not prime
473 is not prime
474 is not prime
475 is not prime
476 is not prime
477 is not prime
478 is not prime
479 is prime
480 is not prime
481 is not prime
482 is not prime
483 is not prime
484 is not prime
485 is not prime
486 is not prime
487 is prime
488 is not prime
489 is not prime
490 is not prime
491 is prime
492 is not prime
493 is not prime
494 is not prime
495 is not prime
496 is not prime
497 is not prime
498 is not prime
499 is prime
500 is not prime
```

الگوریتم

۱. شروع
۲. تعریف لیست خالی res برای ذخیره مقسوم علیه‌ها
۳. اجرای حلقه برای هر number از ۱ تا ۵۰۰:
۴. اجرای حلقه برای هر checker از ۱ تا number
۵. اگر $number \neq 1$ و $number \% checker == 0$ ، checker را به res اضافه کن
۶. در غیر این صورت، حلقه بیرونی را رد کن.
۷. اگر تعداد عناصر res بزرگتر از ۲ باشد:
۸. چاپ کن که number عدد اول نیست.
۹. لیست res را خالی کن.
۱۰. در غیر این صورت:
۱۱. چاپ کن که number عدد اول است.
۱۲. لیست res را خالی کن.
۱۳. پایان

فلوچارت



روش دوم

```
internal class PrimeNumberChecker
{
    0 references
    static void Main(string[] args)
    {
        List<int> res = new List<int>();
        IEnumerable<int> myRange = Enumerable.Range(1, 501);
        for (int number = 1; number <= 500; number++)
        {
            foreach (int checker in myRange)
            {
                if (checker <= number)
                {
                    if (number != 1)
                    {
                        if (number % checker == 0)
                        {
                            res.Add(checker);
                        }
                    }
                    else
                    {
                        goto end_of_outter_loop;
                    }
                }
            }

            if (res.Count > 2)
            {
                Console.Write("{0} is not prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
            else
            {
                Console.Write("{0} is prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
        }
        end_of_outter_loop: { }
        Console.ReadKey();
    }
}
```

روش سوم

```
internal class PrimeNumberChecker
{
    0 references
    static void Main(string[] args)
    {
        List<int> res = new List<int>();
        IEnumerable<int> myRange = Enumerable.Range(1, 500);
        foreach (int number in myRange)
        {
            foreach (int checker in myRange)
            {
                if (checker <= number)
                {
                    if (number != 1)
                    {
                        if (number % checker == 0)
                        {
                            res.Add(checker);
                        }
                    }
                    else
                    {
                        goto end_of_outter_loop;
                    }
                }
            }

            if (res.Count > 2)
            {
                Console.Write("{0} is not prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
            else
            {
                Console.Write("{0} is prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
        }
        end_of_outter_loop: { }
    }
    Console.ReadKey();
}
```


روش چهارم

```
internal class PrimeNumberChecker
{
    0 references
    static void Main(string[] args)
    {
        List<int> res = new List<int>();
        IEnumerable<int> myRange = Enumerable.Range(1, 500);
        foreach (int number in myRange)
        {
            int checker = 1;
            while (checker <= number)
            {
                if (number != 1)
                {
                    if (number % checker == 0)
                    {
                        res.Add(checker);
                        checker++;
                    }
                    else
                    {
                        checker++;
                    }
                }
                else
                {
                    goto end_of_outter_loop;
                }
            }
            if (res.Count > 2)
            {
                Console.Write("{0} is not prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
            else
            {
                Console.Write("{0} is prime", res.Last());
                Console.WriteLine();
                res.Clear();
            }
        }
        end_of_outter_loop: { }
    }
    Console.ReadKey();
}
```

روش ینجم

```
0 references
static void Main(string[] args)
{
    List<int> res = new List<int>();
    int number = 1;
    while (number <= 500)
    {
        int checker = 1;
        while (checker <= number)
        {
            if (number != 1)
            {
                if (number % checker == 0)
                {
                    res.Add(checker);
                    checker++;
                }
                else
                {
                    checker++;
                }
            }
            else
            {
                goto end_of_outter_loop;
            }
        }
        if (res.Count > 2)
        {
            Console.Write("{0} is not prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
        else
        {
            Console.Write("{0} is prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
        end_of_outter_loop: { }
        number++;
    }
    Console.ReadKey();
}
```

روش ششم

```
{
    List<int> res = new List<int>();
    int number = 1;

    while (number <= 500)
    {
        int checker = 1;
        do
        {
            if (number != 1)
            {
                if (number % checker == 0)
                {
                    res.Add(checker);
                    checker++;
                }
                else
                {
                    checker++;
                }
            }
            else
            {
                goto end_of_outter_loop;
            }
        } while (checker <= number);
        if (res.Count > 2)
        {
            Console.Write("{0} is not prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
        else
        {
            Console.Write("{0} is prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
    }
    end_of_outter_loop: { }
    number++;
}
Console.ReadKey();
}
```

روش هفتم

```
static void Main(string[] args)
{
    List<int> res = new List<int>();
    int number = 1;
    do
    {
        int checker = 1;
        do
        {
            if (number != 1)
            {
                if (number % checker == 0)
                {
                    res.Add(checker);
                    checker++;
                }
                else
                {
                    checker++;
                }
            }
            else
            {
                goto end_of_outter_loop;
            }
        }
        while (checker <= number);
        if (res.Count > 2)
        {
            Console.Write("{0} is not prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
        else
        {
            Console.Write("{0} is prime", res.Last());
            Console.WriteLine();
            res.Clear();
        }
    }
    end_of_outter_loop: { }
    number++;
}
while (number <= 500);
Console.ReadKey();
}
```