

به نام خدا

سامان محمدی رئوف ۹۸۵۲۱۴۷۷

پروژه این سوال را به همراه این فایل فرستادم

برای پیاده سازی این سوال از زبان سی شارپ استفاده کردم

همان طور که در سوال گفته کلاس استک و متد های ارزیابی و تبدیل میانوندی به پسوندی را پیاده سازی کردم و برنامه دارای دو قسمت محاسبه نهایی و محاسبه گام به گام می باشد که در محاسبه گام به گام استک ها نمایش داده میشوند در فاصله های زمانی ثابت ۱ ثانیه ای .

در این پروژه دو قسمت وجود دارد یکی فایل مربوط به کد های اصلی و محاسباتی که در واقع بخش مدنظر سوال قرار دارد و یکی هم کد های مربوط به ظاهر اپلیکشن که رابط کاربری هستند .

تمامی کد های محاسباتی و الگوریتم ها نظیر کلاس استک و الگوریتم ارزیابی در فایل Form1.cs هستند . در فایل program.cs هیچ کدی وجود ندارد ، اکسترکت و سپس فایل Form1.cs را برای مشاهده کد باز کنید .

کد کلاس stack

```
public class stack<T>
{
    static public int size = 100;
    public int top = -1;

    public T[] elements = new T[size];
    //constructor dar in ja mana nadarad chon khod barname bayad pishfarz size
    bedahad.
    //public stack(int x )
    //{
    //}

    public T peek()
    {
        if (top == -1)
            throw new EmptyException();
        return elements[top];
    }
    public void Push(T element)
    {
        if (top == size - 1)
        {
            throw new FullStackException();
        }
        top++;
        elements[top]=element;
    }
    public T Pop()
```

```

    {
        T element = peek();
        top--;
        return element;
    }
    public Boolean IsEmpty()
    {
        return -1 == top;
    }
    public Boolean IsFull()
    {
        return top == size - 1;
    }
}

```

کد الگوریتم های ارزیابی و میان وندی به پسوندی

```

public List<string> MianvandToPasvand(string [] array)
{
    stack<string> Operations = new stack<string>();
    for(int i =0; i<array.Length;i++)
    {
        if(array[i].Contains("!"))
        {
            string[] numbers = array[i].Split('!');
            int number = int.Parse(numbers[0]);
            int product = 1;
            for(int j=1;j<=number;j++)
            {
                product = product * j;
            }
            array[i] = product.ToString();
        }
    }

    List<string> answer = new List<string>();
    for (int i = 0; i < array.Length; i++)
    {
        if (array[i] != "+" && array[i] != "-" && array[i] != "*" && array[i] !=
"/" && array[i] != "^")
        {
            answer.Add(array[i]);
        }
        else
        {
            l1:
            if (Operations.top != -1)
            {
                if (Pre(Operations.elements[Operations.top], array[i]))
                {
                    string obj = Operations.Pop();
                    answer.Add(obj.ToString());
                    goto l1;
                }
            }
        }
    }
}

```

```

        }
        else
        {
            Operations.Push(array[i]);
        }
    }

    else
    {
        Operations.Push(array[i]);
    }
}

while(Operations.IsEmpty()==false)
{
    string character = Operations.Pop();
    answer.Add(character);
}
return answer;
}

```

```

public double Arzyabi(List<string> array)
{
    stack<double> mohasebe = new stack<double>();

    for(int i=0;i<array.Count;i++)
    {
        if(array[i]=="+" || array[i] == "-" || array[i] == "*" || array[i] == "/"
|| array[i] == "^" || array[i] == "!")
        {
            double a = mohasebe.Pop();
            double b = mohasebe.Pop();
            double c = 0;
            switch (array[i])
            {
                case "+":
                    c = b + a;
                    break;
                case "-":
                    c = b - a;
                    break;
                case "*":
                    c = b * a;
                    break;
                case "/":
                    c = b / a;

```

```

        break;
    case "^":
        c = Math.Pow(b,a);
        break;
    //case ! be sorat jodagne handle shode.

    }
    mohasebe.Push(c);
}
else
{
    double x = double.Parse(array[i].ToString());
    mohasebe.Push(x);
}
}

double answer = mohasebe.Pop();
return answer;
}

```

## کد الگوریتم pre

```

public bool Pre(string a, string b)
{
    if ((a == "+" || a == "-") && (b == "*" || b == "/" || b == "^" || b == "!"))
    {
        return false;
    }
    else if ((a == "/" || a == "*") && (b == "^" || b == "!"))
    {
        return false;
    }
    else if ((a == "!") && (b == "^"))
    {
        return false;
    }
    else
    {
        return true;
    }
}

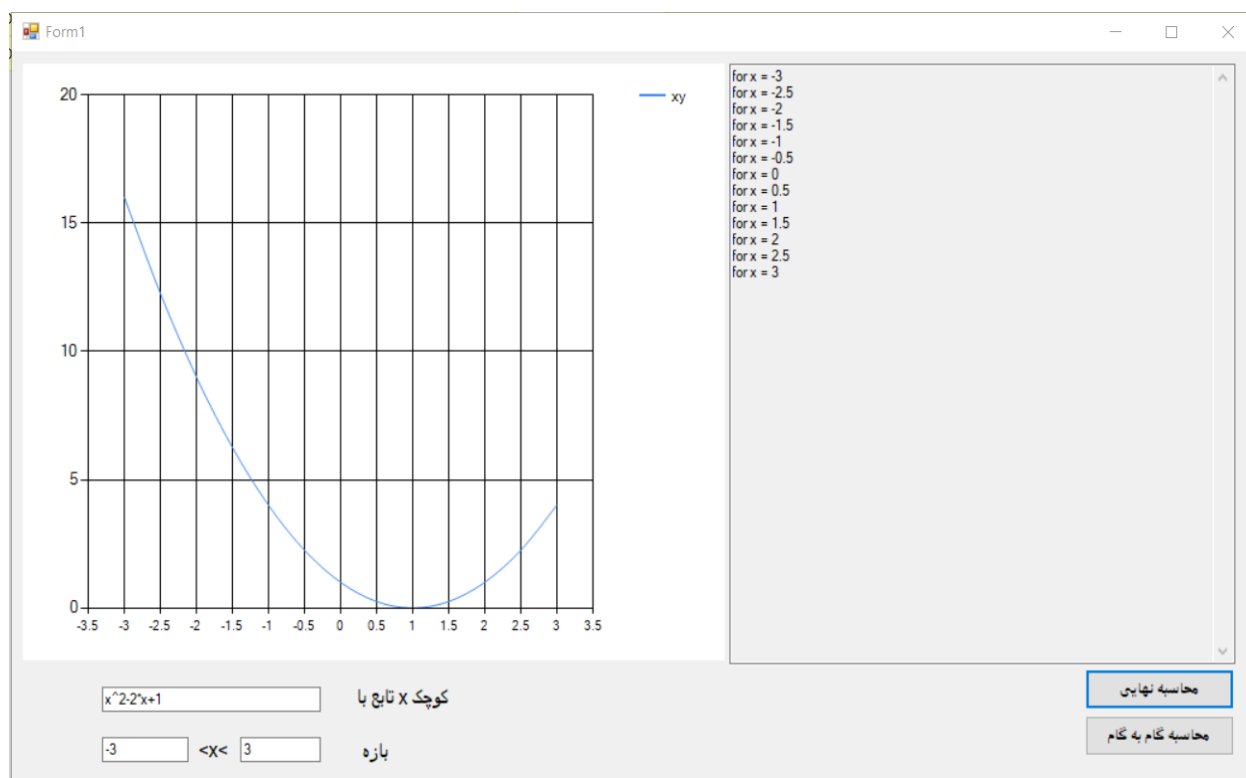
```

البته که این قسمت ها فقط چون در سوال اشاره شده که مهم هستند من اینجا وارد کردم و کد موجود در فایل Form1.cs کامل است

متد های ارزیابی و میانودی به پسوندی هر کدام دارای یک override هستند برای حالتی که میخواهیم گام به گام حساب کنیم تا هر مرحله از این کار ها را با یک تاخیر انجام دهند و نمایش نیز بدهند.

بخش رابط کاربری نیز کد هایش همانطور اشاره کردم در پروژه هست .

خروجی برنامه :  
در حالت محاسبه نهایی



خروجی برنامه در حین محاسبه گام به گام:

در حالت محاسبه گام به گام مراحل در سمت راست نوشته میشوند و در پایان نمودار آن رسم میگردد. (در هنگام محاسبه نمودار رسم نمی شود و باید صبر کرد تا تمام محاسبات انجام شوند تا نمودار کشیده بشود.)

ورودی نیز با  $x$  کوچک تنظیم شده است.

در انتها فایل خروجی برنامه به صورت **exe** در کنار همین فایل هست که قابل اجراست

پایان

با تشکر