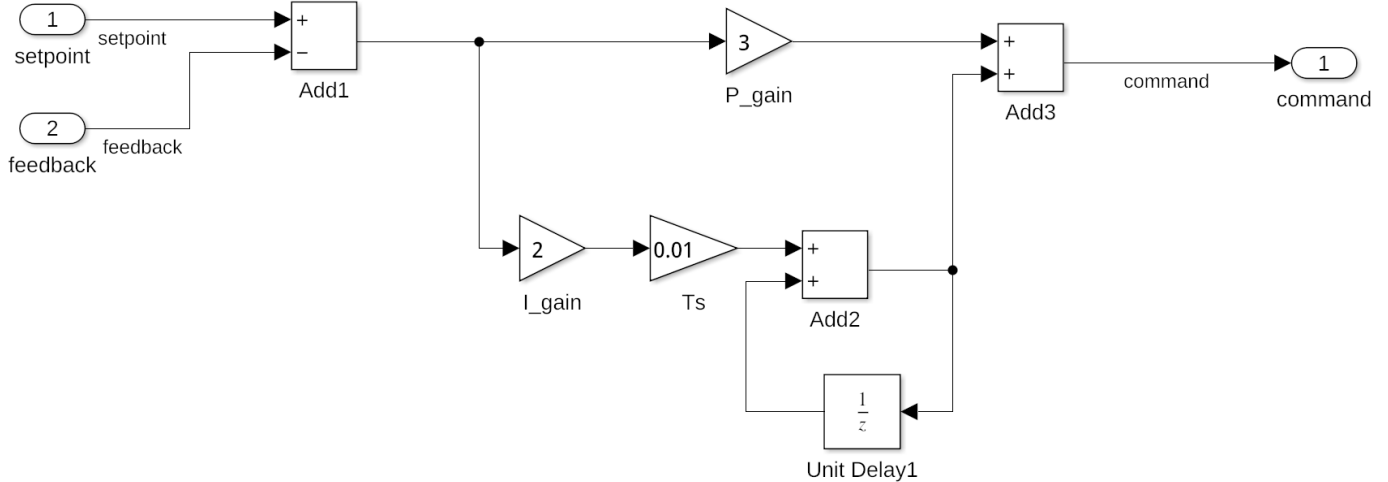


Тестовое задание для собеседования в команду генерации кода

Задан DSL, представляющий собой графическое описание алгоритмов при помощи блок-схем.

Блок-схема представляет из себя блоки со входами и выходами, соединенные направленными линиями. Каждый блок может быть как элементарной операцией (арифметической, логической), так и более сложной составной функцией. Линии обозначают потоки данных между блоками.

Пример блок-схемы, реализующей стандартный ПИ-регулятор:



Формат хранения блок-схемы (в XML):

```
<?xml version="1.0" encoding="utf-8"?>
<System>
  <Block BlockType="Inport" Name="setpoint" SID="16">
    <P Name="Position">[-20, 403, 10, 417]</P>
    <Port>
      <P Name="PortNumber">1</P>
      <P Name="Name">setpoint</P>
    </Port>
  </Block>
  <Block BlockType="Inport" Name="feedback" SID="18">
    <P Name="Position">[-20, 453, 10, 467]</P>
    <P Name="Port">2</P>
    <Port>
      <P Name="PortNumber">1</P>
      <P Name="Name">feedback</P>
    </Port>
  </Block>
  <Block BlockType="Sum" Name="Add1" SID="17">
    <P Name="Ports">[2, 1]</P>
    <P Name="Position">[105, 402, 135, 433]</P>
    <P Name="IconShape">rectangular</P>
    <P Name="Inputs">+-</P>
  </Block>
  <Block BlockType="Sum" Name="Add2" SID="22">
    <P Name="Ports">[2, 1]</P>
    <P Name="Position">[340, 507, 370, 538]</P>
```

```

    <P Name="IconShape">rectangular</P>
</Block>
<Block BlockType="Sum" Name="Add3" SID="23">
    <P Name="Ports">[2, 1]</P>
    <P Name="Position">[430, 412, 460, 443]</P>
    <P Name="IconShape">rectangular</P>
    <Port>
        <P Name="PortNumber">1</P>
        <P Name="Name">command</P>
    </Port>
</Block>
<Block BlockType="Gain" Name="I_gain" SID="25">
    <P Name="Position">[210, 500, 240, 530]</P>
    <P Name="Gain">2</P>
</Block>
<Block BlockType="Gain" Name="P_gain" SID="19">
    <P Name="Position">[305, 405, 335, 435]</P>
    <P Name="Gain">3</P>
</Block>
<Block BlockType="Gain" Name="Ts" SID="26">
    <P Name="Position">[270, 500, 310, 530]</P>
    <P Name="Gain">0.01</P>
</Block>
<Block BlockType="UnitDelay" Name="Unit Delay1" SID="21">
    <P Name="Position">[350, 573, 385, 607]</P>
    <P Name="SampleTime">-1</P>
</Block>
<Block BlockType="Outport" Name="command" SID="20">
    <P Name="Position">[565, 423, 595, 437]</P>
</Block>
<Line>
    <P Name="Name">setpoint</P>
    <P Name="Src">16#out:1</P>
    <P Name="Dst">17#in:1</P>
</Line>
<Line>
    <P Name="Name">feedback</P>
    <P Name="Src">18#out:1</P>
    <P Name="Points">[43, 0; 0, -35]</P>
    <P Name="Dst">17#in:2</P>
</Line>
<Line>
    <P Name="Src">17#out:1</P>
    <P Name="Points">[51, 0]</P>
    <Branch>
        <P Name="Points">[0, 95]</P>
        <P Name="Dst">25#in:1</P>
    </Branch>
    <Branch>
        <P Name="Dst">19#in:1</P>
    </Branch>

```

```

    </Branch>
</Line>
<Line>
  <P Name="Src">21#out:1</P>
  <P Name="Points">[-31, 0; 0, -60]</P>
  <P Name="Dst">22#in:2</P>
</Line>
<Line>
  <P Name="Src">22#out:1</P>
  <P Name="Points">[34, 0]</P>
  <Branch>
    <P Name="Points">[0, 65]</P>
    <P Name="Dst">21#in:1</P>
  </Branch>
  <Branch>
    <P Name="Points">[0, -90]</P>
    <P Name="Dst">23#in:2</P>
  </Branch>
</Line>
<Line>
  <P Name="Src">19#out:1</P>
  <P Name="Dst">23#in:1</P>
</Line>
<Line>
  <P Name="Name">command</P>
  <P Name="Src">23#out:1</P>
  <P Name="Dst">20#in:1</P>
</Line>
<Line>
  <P Name="Src">25#out:1</P>
  <P Name="Dst">26#in:1</P>
</Line>
<Line>
  <P Name="Src">26#out:1</P>
  <P Name="Dst">22#in:1</P>
</Line>
</System>

```

Прямолинейная реализация на Си:

```

#include "nwocg_run.h"

#include <math.h>

static struct
{
  double setpoint;
  double feedback;

```

```

    double Add1;
    double I_gain;
    double Ts;
    double P_gain;
    double UnitDelay1;
    double Add2;
    double Add3;

} nwocg;

void nwocg_generated_init()
{
    nwocg.UnitDelay1 = 0;
}

void nwocg_generated_step()
{
    nwocg.Add1 = nwocg.setpoint - nwocg.feedback;
    nwocg.I_gain = nwocg.Add1 * 2;
    nwocg.Ts = nwocg.I_gain * 0.01;
    nwocg.P_gain = nwocg.Add1 * 3;
    nwocg.Add2 = nwocg.Ts + nwocg.UnitDelay1;
    nwocg.Add3 = nwocg.P_gain + nwocg.Add2;

    nwocg.UnitDelay1 = nwocg.Add2;
}

static const nwocg_ExtPort
ext_ports[] =
{
    { "command", &nwocg.Add3, 0 },
    { "feedback", &nwocg.feedback, 1 },
    { "setpoint", &nwocg.setpoint, 1 },
    { 0, 0, 0 },
};

const nwocg_ExtPort * const
nwocg_generated_ext_ports = ext_ports;

const size_t
nwocg_generated_ext_ports_size = sizeof(ext_ports);

```

Задача:

Разработать прототип генератора Си кода из описанного формата блок-схем.