

LAMPIRAN A
Listing Program

PROGRAM SISTEM PERGERAKKAN WIROBOT X80

```
// RobotDlg.cpp : implementation file
#include "stdafx.h"
#include "Robot.h"
#include "RobotDlg.h"
#include "RobotThread.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
#include "LibDraw.h"
#include <OLEDB.h>

typedef enum {M_PWM, M_POSITION, M_VELOCITY}EControlMode;
#define NO_CONTROL -32768
#define cFULL_COUNT 32767
#define cWHOLE_RANGE 1200
////////////////////////////////////
// CRobotDlg dialog

CRobotDlg::CRobotDlg(CWnd* pParent /*=NULL*/)
: CDialog(CRobotDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CRobotDlg)
    m_InputSpeed = 0;
    m_InputDest = 0;
    m_ShowSpeed = 0;
    m_InputAngle = 0;
    m_EncoderPulse1 = 0;
    m_EncoderPulse2 = 0;
    m_EncoderSpeed1 = 0;
    m_EncoderSpeed2 = 0;
    m_Speed1 = 0;
    m_Speed2 = 0;
   //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_FileName = _T("");
    m_bFinish = false;
    m_MotorTime = 0;
}

void CRobotDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CRobotDlg)
    DDX_Control(pDX, IDC_OUT_ENC2, m_Enc2);
    DDX_Control(pDX, IDC_OUT_ENC1, m_Enc1);
    DDX_Control(pDX, IDC_STAT_FRAME, m_PicFrame);
    DDX_Control(pDX, IDC_EDT_TIME_STOP, m_StopWatch);
    DDX_Control(pDX, IDC_OUT_TIME, m_OutTime);
    DDX_Control(pDX, IDC_EDT_OUTR, m_OutRight);
    DDX_Control(pDX, IDC_EDT_OUTL, m_OutLeft);
    DDX_Text(pDX, IDC_EDT_SPD, m_InputSpeed);
    }}AFX_DATA_MAP
}
```

```

        DDX_Text(pDX, IDC_EDT_DST, m_InputDest);
        DDX_Text(pDX, IDC_ANGL, m_InputAngle);
        DDV_MinMaxInt(pDX, m_InputAngle, -180, 180);
        DDX_Control(pDX, IDC_DRROBOTSDKCONTROLCTRL1, m_WRSDK);
        //}AFX_DATA_MAP
    }
BEGIN_MESSAGE_MAP(CRobotDlg, CDialog)
    //{{AFX_MSG_MAP(CRobotDlg)
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BTN_FWD, OnBtnFwd)
    ON_BN_CLICKED(IDC_BTN_STP, OnBtnStp)
    ON_WM_TIMER()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
// CRobotDlg message handlers

BOOL CRobotDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here
    m_WRSDK.connectRobot("drrobot1");
    CFileException fe;
    char* pFileName = "capture.txt";

    if(!m_File.Open(pFileName, CFile::modeWrite, &fe))
        AfxMessageBox(_T("Couldn't create the file!"));
    else
        m_FileName = m_File.GetFileName();

    m_pDC = m_PicFrame.GetWindowDC();
    m_PicFrame.GetWindowRect(&m_FrameRect);

    InitPic();

    SetTimer(1, 950, NULL);

    UpdateData(FALSE);
    return TRUE; // return TRUE unless you set the focus to a control
}
void CRobotDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting
        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
    }
}

```

```

        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        InitPic();
        CDialog::OnPaint();
    }
}
// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CRobotDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}
BEGIN_EVENTSINK_MAP(CRobotDlg, CDialog)
   //{{AFX_EVENTSINK_MAP(CRobotDlg)
    ON_EVENT(CRobotDlg, IDC_DRROBOTSDKCONTROLCTRL1, 2 /*
MotorSensorEvent */, OnMotorSensorEventDrrobotsdkcontrolctrl1, VTS_NONE)
   //}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

void CRobotDlg::OnMotorSensorEventDrrobotsdkcontrolctrl1()
{
    // TODO: Add your control notification handler code here

    m_EncoderPulse1 = m_WRSDK.GetEncoderPulse1();
    m_EncoderPulse2 = m_WRSDK.GetEncoderPulse2();
    m_EncoderSpeed1 = m_WRSDK.GetEncoderSpeed1();
    m_EncoderSpeed2 = m_WRSDK.GetEncoderSpeed2();
    m_Speed1 = m_EncoderSpeed1 / (short)30;
    m_Speed2 = m_EncoderSpeed2 / (short)30;

    CaptureData();
    ShowOutput();
}

void CRobotDlg::OnBtnFwd()
{
    // TODO: Add your control notification handler code here
    ////////////////////////////////////////////////////////////////////
    // spd          = []
    // m_InputSpeed = [cm/s]
    ////////////////////////////////////////////////////////////////////
    short spd;

    UpdateData(TRUE);

    spd = m_InputSpeed * (short)30;
}

```

```

m_WRSdk.SetDcMotorControlMode(0, M_VELOCITY);
m_WRSdk.SetDcMotorControlMode(1, M_VELOCITY);
m_WRSdk.SetDcMotorVelocityControlPID(0, 100, 0, 0);
m_WRSdk.SetDcMotorVelocityControlPID(1, 48, 0, 48);
t_errcode err = SetAngle();

Sleep(2000);
if(err == SUCCESS)
{
    OnBtnStp();

    m_WRSdk.SetDcMotorControlMode(0, M_VELOCITY);
    m_WRSdk.SetDcMotorControlMode(1, M_VELOCITY);
    m_WRSdk.SetDcMotorVelocityControlPID(0, 100, 0, 0);
    m_WRSdk.SetDcMotorVelocityControlPID(1, 48, 0, 48);
    float v = 0;
    float t = 0;
    m_SecBegin = 0;
    m_SecFinish = 0;
    int milliSecFinish = 0;
    int milliSecBegin = 0;

    v = m_InputSpeed;
    t = m_InputDest / v;
    m_MotorTime = t;

    m_SecBegin = GetSysTimeX(second);
    m_SecFinish = m_SecBegin + t;
    if(m_SecFinish > 60)
        m_SecFinish = m_SecFinish - 60;

    milliSecBegin = GetSysTimeX(msecond);
    if(t < 1)
    {
        milliSecFinish = (t * 100) + milliSecBegin;
        if(milliSecFinish > 999)
            milliSecFinish = milliSecFinish - 999;
    }

    m_WRSdk.DcMotorVelocityNonTimeCtrAll(-spd,
spd, NO_CONTROL, NO_CONTROL, NO_CONTROL, NO_CONTROL);

    AutoBtnAct();
}

void CRobotDlg::OnOK()
{
    // TODO: Add extra validation here
    m_File.Close();
    m_WRSdk.DisableDcMotor (0);
    m_WRSdk.DisableDcMotor (1);

    CDialog::OnOK();
}

```

```

void CRobotDlg::OnBtnStp()
{
    // TODO: Add your control notification handler code here
    m_WRSdk.DisableDcMotor (0);
    m_WRSdk.DisableDcMotor (1);
}

void CRobotDlg::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    int mSecCurrent;
    mSecCurrent = GetSysTimeX(second);

    if(nIDEvent == 1)
    {
        CString sTime = "";

        GetSysTime(sTime);
        m_OutTime.SetSel(0,-1);
        m_OutTime.Clear();
        m_OutTime.ReplaceSel(sTime);
        m_OutTime.UpdateData(false);
    }
    if(nIDEvent == 2)
    {
        if(m_MotorTime > 0)
        {
            CString text = "";
            m_MotorTime--;
            text.Format("%d", m_MotorTime);
            m_StopWatch.SetSel(0,-1);
            m_StopWatch.Clear();
            m_StopWatch.ReplaceSel(text);
            m_StopWatch.UpdateData(false);
            //if(m_SecFinish == mSecCurrent)
            //AutoBtnAct();
        }
        else
            AutoBtnAct();
    }

    CDialog::OnTimer(nIDEvent);
}

void CRobotDlg::GetSysTime(CString &str)
{
    CTime theTime = CTime::GetCurrentTime();
    str = theTime.Format("%A, %B %d, %y - %H:%M:%S");
}

int CRobotDlg::GetSysTimeX(ETime xTime)
{
    int retVal = -1;
}

```

```

    CTime theTime(CTime::GetCurrentTime());
    SYSTEMTIME timeDest;
    theTime.GetAsSystemTime(timeDest);
    switch(xTime)
    {
    case 0:
        retVal = timeDest.wHour;
        break;
    case 1:
        retVal = timeDest.wMinute;
        break;
    case 2:
        retVal = timeDest.wSecond;
        break;
    case 3:
        retVal = timeDest.wMilliseconds;
        break;
    default:
        break;
    }
    return retVal;
}

void CRobotDlg::ShowOutput()
{
    CString strSpd1 = "0";
    CString strSpd2 = "0";
    CString strPuls1 = "0";
    CString strPuls2 = "0";

    strSpd1.Format("%d", m_Speed1);
    strSpd2.Format("%d", m_Speed2);
    strPuls1.Format("%d", m_EncoderPulse1);
    strPuls2.Format("%d", m_EncoderPulse2);

    m_OutLeft.SetSel(0,-1);
    m_OutRight.SetSel(0,-1);
    m_Enc1.SetSel(0,-1);
    m_Enc2.SetSel(0,-1);

    m_OutLeft.Clear();
    m_OutRight.Clear();
    m_Enc1.Clear();
    m_Enc2.Clear();

    m_OutLeft.ReplaceSel(strSpd1);
    m_OutRight.ReplaceSel(strSpd2);
    m_Enc1.ReplaceSel(strPuls1);
    m_Enc2.ReplaceSel(strPuls2);

    m_OutLeft.UpdateData(false);
    m_OutRight.UpdateData(false);
    m_Enc1.UpdateData(false);
    m_Enc2.UpdateData(false);
}

```

```

        m_OutEnc1 = atol(strPuls1);
        m_OutEnc2 = atol(strPuls2);
    }

void CRobotDlg::CaptureData()
{
    CString strText, strTextLeft, strTextRight, strTextEnc1, strTextEnc2;
    int lenTxtLeft, lenTxtRight;
    int lenTxtEnc1, lenTxtEnc2;
    CString TextSpeed = "";
    static const char sz[] = "=====\n";

    int len = GetText(&m_OutTime, strText);

    m_File.SeekToEnd();
    m_File.Write(strText + "\n", len+1);
    m_File.Write(sz, lstrlen(sz));

    lenTxtLeft = GetText(&m_OutLeft, strTextLeft);
    lenTxtRight = GetText(&m_OutRight, strTextRight);
    lenTxtEnc1 = GetText(&m_Enc1, strTextEnc1);
    lenTxtEnc2 = GetText(&m_Enc2, strTextEnc2);

    m_File.Write(strTextLeft + "|" + strTextRight + "\t" + strTextEnc1 + "|" + strTextEnc2 +
"\n", lenTxtLeft + lenTxtRight + lenTxtEnc1 + lenTxtEnc2 + 3);
}

int CRobotDlg::GetText(CEdit *pEditText, CString& text)
{
    int len = pEditText->LineLength(pEditText->LineIndex(0));
    pEditText->GetLine(0, text.GetBuffer(len), len);
    text.ReleaseBuffer(len);
    return len;
}

void CRobotDlg::AutoBtnAct()
{
    if(m_bFinish == false)
    {
        m_bFinish = true;
        SetTimer(2, 950, NULL);
    }
    else
    {
        KillTimer(2);
        // disable DC Motor
        m_WRSdk.DisableDcMotor (0);
        m_WRSdk.DisableDcMotor (1);
        m_bFinish = false;
    }
}

t_errcode CRobotDlg::SetAngle()

```



```

{
    int retVal = FAILURE;
    int absAngle;

    absAngle = abs(m_InputAngle);
    if(absAngle > 180)
    {
        // return failure input angle value must be int from 0 to 180
        return retVal;
    }
    else
    {
        bool signAngle;
        CMFCRobotImg img;
        img.Init(m_pDC);
        img.Set(m_X,m_Y);

        signAngle = m_InputAngle > 0 ? true : false;
        if(signAngle == true)
        {
            // turn left

            long cmd1,cmd2;

            short angl = absAngle * 5.11;
            cmd1 = m_OutEnc1 - angl;
            cmd2 = m_OutEnc2 - angl;

            //change cmd1, cmd2 to valid data range
            if (cmd1 < 0) cmd1 = cmd1 + cFULL_COUNT;
            if (cmd2 < 0) cmd2 = cmd2 + cFULL_COUNT;
            if (cmd1 > cFULL_COUNT) cmd1 = cmd1 - cFULL_COUNT;
            if (cmd2 > cFULL_COUNT) cmd2 = cmd2 - cFULL_COUNT;

            m_WRSdk.SetDcMotorControlMode (0,M_POSITION);
            m_WRSdk.SetDcMotorControlMode (1,M_POSITION);

            m_WRSdk.SetDcMotorPositionControlPID (0, 600, 30, 600);
            m_WRSdk.SetDcMotorPositionControlPID (1, 600, 30, 600);

            m_WRSdk.DcMotorPositionTimeCtrAll
(cmd1,cmd2,NO_CONTROL,NO_CONTROL,NO_CONTROL,NO_CONTROL,1000);

            retVal = SUCCESS;
        }
        else
        {
            // turn right
            long cmd1,cmd2;
            short angl = absAngle * 5.11;
            cmd1 = m_OutEnc1 + angl;
            cmd2 = m_OutEnc2 + angl;

```

```

        //change cmd1, cmd2 to valid data range
        if (cmd1 < 0) cmd1 = cmd1 + cFULL_COUNT;
        if (cmd2 < 0) cmd2 = cmd2 + cFULL_COUNT;
        if (cmd1 > cFULL_COUNT) cmd1 = cmd1 - cFULL_COUNT;
        if (cmd2 > cFULL_COUNT) cmd2 = cmd2 - cFULL_COUNT;

        m_WRSdk.SetDcMotorControlMode (0,M_POSITION);
        m_WRSdk.SetDcMotorControlMode (1,M_POSITION);

        m_WRSdk.SetDcMotorPositionControlPID (0, 600, 30, 600);
        m_WRSdk.SetDcMotorPositionControlPID (1, 600, 30, 600);
        m_WRSdk.DcMotorPositionTimeCtrAll
(cmd1,cmd2,NO_CONTROL,NO_CONTROL,NO_CONTROL,NO_CONTROL,1000);
        retVal = SUCCESS;
    }
}
return retVal;
}
void CRobotDlg::InitPic()
{
    int x1, x2, y1, y2, r, width, height;

    width = m_FrameRect.right-m_FrameRect.left;
    height = m_FrameRect.bottom-m_FrameRect.top;

    m_X = x1 = abs(width/2);
    m_Y = y1 = abs(height/2);

    x2 = x1 + width;
    y2 = y1 + height;
    r = x1-28;

    CMFCRadar newRadar(x1,y1,r);

    newRadar.SetBorderStyle(0xffffffff);
    newRadar.SetBorderColor(RGB(0,0,255));
    newRadar.SetBorderWidth(2);
    newRadar.SetBorder(true);
    newRadar.Draw(m_pDC);

    CMFCRobotImg theImg;
    theImg.Init(m_pDC);
    theImg.Set(x1,y1);
    theImg.SetAngle(0);
}

```

LAMPIRAN B
TABEL DATA PERCOBAAN

Tabel 1. Kecepatan Motor Kanan dengan beberapa Parameter PID

Detik ke-	Kecepatan motor kanan							
	Kp=13.32 Ki=0 Kd=0	Kp=20 Ki=0 Kd=0	Kp=30 Ki=0 Kd=0	Kp=40 Ki=0 Kd=0	Kp=48 Ki=0 Kd=0	Kp=48, Ki=4.8 Kd=0	Kp=48 Ki=0.48 Kd=0	
1	0	0	0	0	0	0	0	
	20	11	30	24	34	16	22	
	20	20	25	28	23	30	21	
	18	19	20	22	21	23	19	
	17	17	19	20	19	20	20	
	15	18	19	19	21	23	20	
	20	18	20	19	18	16	20	
	18	17	21	20	20	22	20	
	18	18	19	18	17	19	20	
	17	17	20	19	21	22	21	
	18	19	18	19	19	19	20	
	18	18	17	16	21	21	20	
	15	18	20	22	19	20	21	
	18	18	19	16	20	17	22	
	17	19	17	22	20	24	20	
2	16	18	20	18	22	20	19	
	18	18	19	20	20	19	19	
	18	17	20	19	19	22	20	
	16	19	20	20	19	20	20	
	15	17	18	20	19	18	20	
	19	19	19	20	20	18	19	
	19	16	19	20	20	20	19	
	17	17	19	20	22	20	18	
	18	16	19	19	24	19	19	
	18	15	19	19	20	20	19	
	16	20	18	20	20	15	19	
	17	17	18	19	18	25	20	
	17	17	20	24	19	19	20	
	20	17	20	20	21	19	21	
	17	18	20	19	20	18	20	
3	17	19	18	18	20	20	20	
	17	17	21	19	20	20	21	
	20	18	19	20	20	18	21	
	19	19	20	17	20	20	21	
	19	18	20	21	18	20	20	
	17	18	19	18	23	23	19	
	18	17	18	20	20	20	19	
	17	18	18	18	20	18	20	
	17	17	18	20	20	20	20	
	17	18	18	18	20	23	20	
	4	18	18	19	22	18	20	21
		18	17	20	21	20	16	20
		16	17	22	20	20	20	19
		19	17	19	19	24	17	21
		16	17	18	18	21	20	22
16		15	18	20	20	17	20	
17		21	19	18	20	20	20	
13		19	21	20	24	18	20	
19		17	18	21	20	20	21	

5	17	18	21	20	22	20	21
	15	18	18	17	19	17	21
	19	20	20	21	22	23	20
	18	18	18	19	19	20	20
	17	17	20	19	17	20	22
	15	20	20	20	24	19	20
	11	18	19	18	20	20	20
	5	17	19	20	18	16	19
	2	13	14	16	15	12	17
	0	10	11	13	9	10	13
6		7	8	10	6	3	12
		5	6	7	4	0	9
		2	1	4	1		3
		0	0	2	0		0
				0			
				0			

Tabel 2. Data dengan Parameter PID yang berbeda untuk masing-masing motor

Detik ke- (s)	PARAMETER PID KIRI (L) & KANAN (R)									
	L	R	L	R	L	R	L	R	L	R
	Kp=15 Ki=0 Kd=0	Kp=48 Ki=0.48 Kd=0	Kp=45 Ki=0 Kd=0	Kp=48 Ki=0.48 Kd=0	Kp=70 Ki=0 Kd=0	Kp=48 Ki=0.48 Kd=0	Kp=100 Ki=0 Kd=0	Kp=48 Ki=0.48 Kd=0	Kp=100 Ki=0.5 Kd=0	Kp=48 Ki=0.48 Kd=0
KECEPATAN MOTOR KIRI (L) DAN KANAN (R)										
	L	R	L	R	L	R	L	R	L	R
1	0	0	0	0	0	0	0	0	0	0
	25	22	19	19	17	16	19	17	9	6
	22	21	19	19	21	20	20	20	19	17
	20	20	21	21	20	19	20	18	20	20
	19	20	20	20	20	20	20	20	19	18
	18	20	18	19	20	19	20	20	21	20
	17	22	20	20	17	19	20	19	20	18
	18	22	20	20	20	21	20	20	18	22
	17	21	19	20	20	20	18	18	21	18
	18	21	20	19	20	20	20	20	20	21
	17	20	18	20	18	20	19	19	20	21
	17	20	19	20	20	20	20	20	20	21
	17	20	18	19	18	21	20	19	20	19
	16	21	19	20	23	21	18	20	19	19
	16	21	20	20	20	21	21	20	20	20
	17	22	20	20	20	20	20	21	20	19
	18	20	20	20	20	19	19	20	18	20
17	20	19	19	16	20	20	20	21	18	
18	20	20	20	20	19	20	20	18	21	
2	17	21	19	20	19	19	20	19	21	19
	14	20	21	20	20	20	21	20	20	20
	21	22	19	20	18	19	20	19	20	19
	28	21	18	21	20	20	20	20	20	19
	26	21	20	20	18	19	19	20	22	18
	22	20	17	20	21	21	20	20	18	21
	20	19	21	20	19	21	20	19	20	21
	19	21	20	20	20	20	20	20	19	21
	17	21	20	19	14	20	21	19	20	19

3	17	21	18	20	21	20	20	20	20	21
	17	20	20	19	20	19	21	20	21	21
	17	19	19	20	20	19	20	19	19	21
	17	20	20	20	20	20	20	21	20	20
	17	20	18	20	20	20	19	18	19	21
	17	21	20	22	20	22	20	21	22	19
	17	20	19	20	18	20	20	20	18	20
	17	21	18	19	19	20	20	19	20	19
	17	21	19	20	19	20	20	20	20	19
	17	21	19	19	20	19	21	20	20	19
4	18	20	19	20	19	20	19	20	19	20
	17	17	19	18	19	19	19	19	20	19
	1	21	20	20	18	20	18	20	20	20
	28	19	20	19	20	20	20	18	19	20
	35	19	18	19	20	20	20	20	21	20
	31	19	19	20	20	20	18	20	21	19
	26	22	19	20	21	20	21	19	20	21
	23	22	19	20	20	20	21	20	18	20
	20	19	20	19	20	19	21	18	20	19
	18	21	18	20	20	21	21	20	18	20
5	17	20	21	20	20	20	20	19	21	19
	17	19	18	18	19	20	20	19	21	20
	16	20	20	19	19	19	20	19	21	18
	17	18	20	20	20	20	19	19	22	20
	17	20	17	21	20	20	20	20	19	18
	17	20	20	22	20	20	20	20	20	20
	17	19	17	21	23	19	20	20	20	19
	18	20	20	21	17	20	20	20	20	21
	15	18	19	19	15	14	16	14	21	20
	12	15	16	19	13	14	14	12	20	19
6	9	11	15	19	10	10	11	10	17	15
	6	7	13	16	9	9	10	8	15	10
	0	5	11	13	6	6	7	5	11	8
	0	4	10	11	5	5	5	4	9	5
	0	1	7	8	2	1	3	0	7	4
	0	0	5	6	0	0	0	0	5	2
			4	5					4	0
			2	2					2	0
			0	0					0	0

Tabel 3. Percobaan dengan beberapa set point kecepatan
 Parameter motor kiri : $K_p = 100$, $K_i = 0$, $K_d = 0$
 Parameter motor kanan : $K_p = 48$, $K_i = 0.48$, $K_d = 0$

Detik ke- (s)	Set point 10 cm/s				Set point 20 cm/s				Set point 30 cm/s				Set point 40 cm/s				Set point 50 cm/s			
	I		II		I		II		I		II		I		II		I		II	
	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	12	13	12	9	8	4	3	21	21	15	18	23	20	12	13	23	21	20	20	20
10	11	9	13	18	17	20	16	23	24	21	21	25	25	23	23	25	25	24	22	22
10	11	12	10	21	21	20	20	29	30	30	29	33	32	27	28	31	31	30	29	29
10	9	9	10	20	20	19	18	31	31	31	29	35	35	33	33	35	35	34	33	33

1	10	10	10	10	20	20	20	21	28	29	27	30	36	35	35	34	36	38	35	35
	10	10	12	10	20	20	20	20	28	28	30	29	38	36	35	35	35	36	38	35
	10	10	10	10	20	20	20	20	30	30	29	29	37	35	37	35	36	39	38	38
	10	10	10	10	20	21	18	19	28	28	29	30	37	35	37	35	36	36	39	39
	10	10	9	9	19	21	20	20	28	27	28	30	37	36	39	40	37	35	39	40
	11	11	10	10	22	21	20	20	29	30	29	29	38	35	37	34	38	36	39	38
	10	10	10	10	20	19	20	20	28	30	30	30	36	35	38	31	38	35	39	39
	10	11	10	10	20	19	21	19	29	29	29	29	37	35	35	42	40	36	41	40
	10	9	11	10	20	20	20	19	28	29	29	29	37	35	35	34	40	38	40	39
	10	10	10	10	19	21	20	20	28	28	30	31	37	35	37	35	40	39	40	37
2	10	10	10	10	20	20	19	19	30	30	29	29	36	35	36	33	40	39	38	35
	10	10	10	10	19	19	21	21	27	27	28	28	36	35	35	34	40	39	40	38
	10	10	10	10	20	21	19	20	30	29	30	28	38	39	37	36	40	40	40	39
	11	10	10	10	20	20	20	22	27	27	29	29	37	34	37	35	38	40	40	40
	10	12	10	10	20	20	19	19	28	28	31	31	36	35	38	38	39	38	41	39
	11	11	10	10	19	19	20	20	30	28	29	29	36	35	37	35	40	37	41	39
	11	11	9	10	20	20	20	20	28	29	30	29	36	35	38	35	40	37	40	34
	11	10	11	10	18	19	20	19	28	29	28	29	37	34	36	37	40	37	40	38
	10	10	10	9	21	20	19	20	27	27	29	29	37	36	37	37	42	40	39	36
	10	10	10	11	19	21	20	17	29	28	27	28	37	36	36	35	40	39	40	38
3	10	10	10	10	21	17	19	21	26	28	30	30	36	37	39	35	40	40	39	36
	10	10	10	10	18	23	20	20	30	29	28	29	36	37	36	33	40	39	39	37
	10	10	10	10	21	19	20	20	29	28	29	29	36	35	35	35	36	37	39	36
	10	10	11	10	20	22	20	20	28	29	28	30	37	37	37	35	38	39	38	37
	10	10	11	10	20	18	20	20	30	29	30	27	37	35	36	33	40	37	40	38
	10	10	10	10	20	20	21	19	27	27	27	29	36	35	37	36	39	36	40	36
	11	11	12	10	20	20	20	21	28	28	30	29	38	35	39	37	39	36	40	37
	9	10	10	11	20	20	19	19	29	28	27	29	38	36	36	35	40	37	40	38
	11	10	10	12	20	20	20	20	28	27	29	28	36	35	37	37	41	40	40	40
	11	11	10	10	20	19	20	20	29	27	28	27	38	37	37	37	40	37	40	39
4	10	10	11	11	20	20	20	21	26	27	30	28	37	37	35	35	40	40	40	37
	11	11	11	10	20	20	20	20	29	28	28	28	37	37	35	35	40	40	39	35
	10	9	10	10	21	19	19	19	28	27	28	27	36	35	35	35	40	40	40	38
	10	10	10	10	19	20	21	20	25	25	27	29	37	37	35	35	40	40	39	38
	10	10	10	9	20	19	19	21	24	24	23	25	35	35	35	35	40	40	40	39
	10	10	9	10	19	20	20	21	22	23	23	21	36	35	36	35	40	38	40	38
	10	10	12	9	20	20	20	20	20	20	20	19	37	36	37	36	42	37	40	37
	10	9	9	12	20	20	20	20	19	20	20	18	37	35	37	35	42	41	39	35
	9	10	11	9	19	20	20	20	15	16	15	16	37	34	37	34	41	39	40	37
	11	10	10	10	20	19	20	20	14	15	10	11	36	36	36	36	41	38	38	37
5	11	10	10	10	18	18	21	20	10	11	8	9	36	36	36	36	40	40	27	32
	10	11	10	10	20	20	20	18	10	10	5	5	36	36	36	36	40	39	22	30
	10	9	10	10	20	21	21	20	6	5	3	4	38	35	38	35	33	26	18	27
	10	10	10	10	20	20	19	19	5	5	0	1	37	35	37	35	31	20	16	27
	11	10	10	10	21	16	21	21	2	3	0	0	37	35	37	35	31	15	15	22
	10	10	10	10	16	20	18	19	0	0			39	35	39	35	28	14	13	20
	10	10	10	9	20	20	20	20					36	36	36	36	25	10	11	18
	9	10	10	10	20	20	20	20					36	37	37	38	22	9	10	16
	11	9	9	10	21	19	20	20					36	34	38	36	20	5	8	14
	10	10	11	10	20	20	21	21					36	35	36	36	16	3	6	12
11	10	10	10	20	20	20	19					37	36	37	37	14	2	3	8	
9	10	10	10	19	21	20	20					37	35	36	36	13	0	1	6	
11	9	10	10	18	19	18	18					38	35	33	37	10	0	0	3	
11	10	10	10	18	18	16	16					37	38	30	30	8	0	0	1	
11	10	10	10	16	17	15	15					37	35	25	28	5	0	0	0	

