

<Revision 정보>

일자	VERSION	변경내역	작성자
2007. 03. 27	0.1	초기 작성	박준완

데스크탑 솔루션 개발지원 보고서

KDEEdu Solution(Kbruch)

기능 개선 및 한글화 작업

한국소프트웨어진흥원
공개SW기술지원센터

목 차

1. 문서 개요	4
가. 문서의 목적	4
나. 본 문서의 사용방법	4
다. 참고 사항	4
2. 개발지원 내역	5
가. 개발 지원 이유	5
나. 개발 사항	5
3. Kbruch 메뉴얼	14
가. 소개글	14
나. Kbruch 사용법	14
다. 연습	15
라. 만듦사람 및 License	17
마. 부록A. 설치 방법	18

< 그림 차례 >

1. Kbruch Original ScreenShot	5
2. Kbruch 한글화 작업 1차 완료 ScreenShot	6
3. KBruch Main 화면	27
4. 화면 설정	28
5. 분수 연습 화면	30
6. 에러 메시지	31
7. 비교연습 화면	31
8. 변환연습 화면	33
9. 인수분해 연습 화면	33

< 표 차례 >

1. Kbruch Source Tree 구조	6
--------------------------------	---

1. 문서 개요

본 문서는 KIPA 공개S/W 기술지원 센터에서의 리눅스 데스크탑 솔루션 개발지원 결과를 보고하기 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- 0 KDEEdu 솔루션의 국내 리눅스 OS 탑재 및 한글화, 메뉴얼 작업을 통한 활용 편의성 증대
- 0 Kbruch로 학생들의 수학 실력을 향상 시킬 수 있으며 수학에 흥미를 유도할 수 있다.

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- 0 수학시간에 학생들의 수업 참여도를 향상 시키며 수학에 흥미를 유도할 수 있다 .
- 0 개발 지원 결과 내역을 참조한다.

다. 참고사항

- 0 개발지원시 시스템 관련사항, 기술적 배경 등 참고사항들을 기술한다.

2. 개발지원 내용

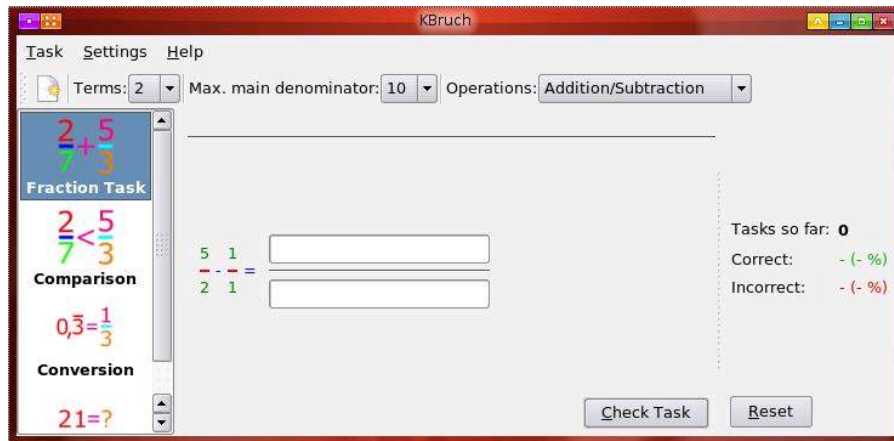
1. 개발 지원 이유

- 가. KDE 커뮤니티에서 만드는 KDEEdu Solution은 다양한 프로그램과 고급 기능, 친숙한 인터페이스와 쉬운 사용법등 여러가지 장점을 가진 교육용 프로그램이다. 이 프로그램들은 초, 중, 고등학교 학생들이 좀더 친숙하게 수학, 언어, 과학등을 접할 수 있게끔 구성되어 있다.
- 이 프로그램들이 영어로 되어 있기 때문에 어휘력이 부족한 학생들에게는 어려운 프로그램으로 인식 될 수 있다. 이에 프로그램을 한글화 하며, 부족한 기능을 추가, 개선하고 메뉴얼 작성을 통해 학생들이 손쉽게 접하여 사용할 수 있도록 하는데 목적이 있다.

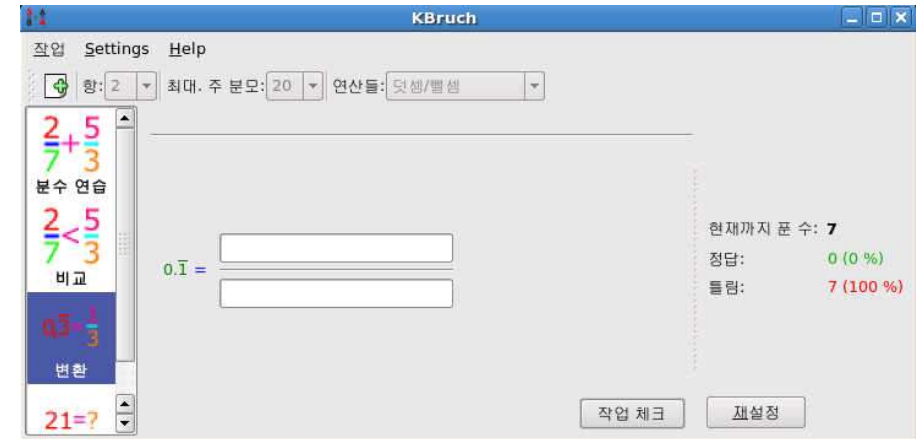
2. 개발 사항

개발 사항은 한글화 작업, 메뉴얼과 기능개선 작업으로 구분되어 있다.

가. 한글화 작업



<그림 1> Kbruch Original ScreenShot



<그림 2> Kbruch 한글화 작업 1차 완료 ScreenShot

한글화 작업은 화면에 보이는 부분과 마우스 이벤트를 통한 설명 부분 모두 진행하였으며, 전체 큰 틀(메뉴)부분은 2차 작업을 통해 진행될 예정이다.

※ Kbruch Source Tree 구조 <표 1>

```
kbruch - AUTHORS
- ChangeLog
- Makefile
- Makefile.am
- Makefile.in
- NEWS
- README
- TODO
- VERSION
- kbruch.lsm
- testcases - Makefile
- Makefile.am
- Makefile.in
- README
- kbruch_test.cpp
- primenumber_test.cpp
- ratio_test.cpp
```

- src - Makefile
- Makefile.am
- Makefile.in
- cr32-action-kbruch_exercise_common.png
- cr32-action-kbruch_exercise_compare.png
- cr32-action-kbruch_exercise_conversion.png
- cr32-action-kbruch_exercise_factorisation.png
- cr64-action-kbruch_exercise_common.png
- cr64-action-kbruch_exercise_compare.png
- cr64-action-kbruch_exercise_conversion.png
- cr64-action-kbruch_exercise_factorisation.png
- exercisebase.cpp
- exercisebase.h
- exercisebase.moc
- exercisebase.o
- exercisecompare.cpp
- exercisecompare.h
- exercisecompare.moc
- exercisecompare.o
- exerciseconvert.cpp
- exerciseconvert.h
- exerciseconvert.moc
- exerciseconvert.o
- exercisefactorize.cpp
- exercisefactorize.h
- exercisefactorize.moc
- exercisefactorize.o
- factorizedwidget.cpp
- factorizedwidget.h
- factorizedwidget.moc
- factorizedwidget.o
- fractionbasewidget.cpp
- fractionbasewidget.h
- fractionbasewidget.moc
- fractionbasewidget.o
- hi128-app-kbruch.png
- hi16-app-kbruch.png
- hi22-app-kbruch.png
- hi32-app-kbruch.png
- hi48-app-kbruch.png
- hi64-app-kbruch.png
- hisc-app-kbruch.svgz

- kbruch
- kbruch.cpp
- kbruch.desktop
- kbruch.h
- kbruch.kcfg
- kbruch.o
- kbruchui.rc
- mainqtwidget.cpp
- mainqtwidget.h
- mainqtwidget.moc
- mainqtwidget.o
- primenumber.cpp
- primenumber.h
- primenumber.o
- ratio.cpp
- ratio.h
- ratio.o
- rationalwidget.cpp
- rationalwidget.h
- rationalwidget.moc
- rationalwidget.o
- ratiowidget.cpp
- ratiowidget.h
- ratiowidget.moc
- ratiowidget.o
- resultwidget.cpp
- resultwidget.h
- resultwidget.moc
- resultwidget.o
- settingsclass.cpp
- settingsclass.h
- settingsclass.kcfgc
- settingsclass.o
- statisticsview.cpp
- statisticsview.h
- statisticsview.moc
- statisticsview.o
- task.cpp
- task.h
- task.o
- taskview.cpp
- taskview.h

```
- taskview.moc
- taskview.o
- taskvieweroptionsbase.cpp
- taskvieweroptionsbase.h
- taskvieweroptionsbase.moc
- taskvieweroptionsbase.o
- taskvieweroptionsbase.ui
- taskwidget.cpp
- taskwidget.h
- taskwidget.moc
- taskwidget.o
- version.h
- svg-source - cr-action-kbruch_exercise_common.svg
               - cr-action-kbruch_exercise_conversion.svg
               - cr-action-kbruch_exercise_compare.svg
               - cr-action-kbruch_exercise_factorisation.svg
```

- ※ PNG 파일은 솔루션 내 그림파일이다.
- ※ Makefile은 솔루션 개별적으로 make 가 가능하도록 만들어져 있는 파일이다.
- ※ .o 파일은 make후 만들어진 object 파일이다.

< 한글화 작업 File Source Sample>

※ mainqtwidget.cpp (메인 윈도우 관련 파일)

```

//*****
mainqtwidget.cpp - The main Qt/KDE window
-----
begin : Tue Mar 16 00:00:00 CET 2003
copyright : (C) 2003-2004 by Sebastian Stein
email : seb.kde@hpfs.de
*****/
/*****
*
* This program is free software; you can redistribute it and/or modify *
* it under the terms of the GNU General Public License as published by *
* the Free Software Foundation; either version 2 of the License, or *
* (at your option) any later version. *
*
*****/
#include "mainqtwidget.h"
```

```

#include <config.h>
#include <kaccel.h>
#include <kapplication.h>
#include <kaction.h>
#include <kdebug.h>
#include <kiconloader.h>
#include <kjanuswidget.h>
#include <kconfigdialog.h>
#include <klocale.h>
#include <qspliter.h>
#include <qlabel.h>
#include <qtooltip.h>
#include <qwhatsthis.h>
#include <qwidget.h>
#include <math.h>
#include "exercisecompare.h"
#include "exerciseconvert.h"
#include "exercisefactorize.h"
#include "taskview.h"
#include "taskvieweroptionsbase.h"
#include "statisticsview.h"
#include "settingsclass.h"
/* ----- public member functions ----- */
MainQtWidget::MainQtWidget()
{
#ifdef DEBUG
kdDebug() << "constructor MainQtWidget" << endl;
#endif
// get the settings
readOptions();
// creating KActions, used by the kbruchui.rc file
setupActions();
createGUI(0L);
// we split the main view into 2 parts, one for the tasks, one for the
// statistics
QSplitter* splitter = new QSplitter(QSplitter::Horizontal, this, "QSplitter");
setCentralWidget(splitter);
// the iconlist, where the user can choose the different exercises
m_exercises = new KJanusWidget(splitter, "KJanusWidget", KJanusWidget::IconList);
QToolTip::add(m_exercises, i18n("아이콘을 누르면 다른 연습을 선택할 수 있습니다."));
QWhatsThis::add(m_exercises, i18n("다른 아이콘을 누르면 다른 연습을 선택할 수 있습니다. 이
연습들은 함수를 계산하는 여러가지 다른 방식들을 연습할 수 있습니다."));
// create the statistic view
```

```

m_statview = new StatisticsView(splitter, "StatisticsView");
// add the pages
//
// we have the exercise to solve fraction tasks
QVBoxLayout * page = m_exercises->addVBoxLayout(i18n("분수 연습"), "",
DesktopIcon("kbruch_exercise_common"));
m_taskview = new TaskView((QWidget *) page, "TaskView", m_addSub, m_mulDiv,
m_nrRatios, m_maxMainDenominator);
// we have the exercise to compare ratios
page = m_exercises->addVBoxLayout(i18n("비교"), "",
DesktopIcon("kbruch_exercise_compare"));
m_exerciseCompare = new ExerciseCompare((QWidget *) page, "ExerciseCompare");
// we have the exercise to convert rational numbers into ratios
page = m_exercises->addVBoxLayout(i18n("변환"), "",
DesktopIcon("kbruch_exercise_conversion"));
m_exerciseConvert = new ExerciseConvert((QWidget *) page, "ExerciseConvert");
// we have the exercise to factorize a given number into prime factors
page = m_exercises->addVBoxLayout(i18n("인수분해"), "",
DesktopIcon("kbruch_exercise_factorisation"));
m_exerciseFactorize = new ExerciseFactorize((QWidget *) page, "ExerciseFactorize");
splitter->setResizeMode(m_statview, QSplitter::FollowSizeHint);
// we must change the status of the menubar before another page is shown
QObject::connect(m_exercises, SIGNAL(aboutToShowPage(QWidget *)), this,
SLOT(slotAboutToShowPage(QWidget *)));
// connect signals of the exercises and StatisticView, so that StatisticView
// gets informed about how the user solved a given task (wrong or correct)
QObject::connect(m_taskview, SIGNAL(signalTaskSolvedCorrect()), m_statview,
SLOT(addCorrect()));
QObject::connect(m_taskview, SIGNAL(signalTaskSolvedWrong()), m_statview,
SLOT(addWrong()));
QObject::connect(m_exerciseCompare, SIGNAL(signalExerciseSolvedCorrect()), m_statview,
SLOT(addCorrect()));
QObject::connect(m_exerciseCompare, SIGNAL(signalExerciseSolvedWrong()), m_statview,
SLOT(addWrong()));
QObject::connect(m_exerciseConvert, SIGNAL(signalExerciseSolvedCorrect()), m_statview,
SLOT(addCorrect()));
QObject::connect(m_exerciseConvert, SIGNAL(signalExerciseSolvedWrong()), m_statview,
SLOT(addWrong()));
QObject::connect(m_exerciseFactorize, SIGNAL(signalExerciseSolvedCorrect()), m_statview,
SLOT(addCorrect()));
QObject::connect(m_exerciseFactorize, SIGNAL(signalExerciseSolvedWrong()), m_statview,
SLOT(addWrong()));

```

```

#if (KDE_VERSION_MINOR>=3) && (KDE_VERSION_MAJOR>=3)
#else
resize(QSize(QMAX(toolBar()->sizeHint().width(), sizeHint().width()), sizeHint().height()));
#endif
// now show the last exercise
m_exercises->showPage(SettingsClass::activeExercise());
slotAboutToShowPage(m_exercises->pageWidget(m_exercises->activePageIndex()));
}MainQtWidget::~MainQtWidget()
{ /* ----- private member functions ----- */
void MainQtWidget::readOptions()
{ m_addSub = SettingsClass::addsub();
m_mulDiv = SettingsClass::muldiv();
m_nrRatios = SettingsClass::number_ratios();
m_maxMainDenominator = SettingsClass::max_main_denominator();
/* make sure that we can load config files with corrupted values */
if (m_mulDiv == true && pow(2, m_nrRatios) > m_maxMainDenominator)
{
m_nrRatios = 2;
m_maxMainDenominator = 10;
}
}void MainQtWidget::writeOptions()
{ SettingsClass::setActiveExercise(m_exercises->activePageIndex());
// save settings for exercise solve task with fractions
SettingsClass::setAddsub(m_addSub);
SettingsClass::setMuldiv(m_mulDiv);
SettingsClass::setNumber_ratios(m_nrRatios);
SettingsClass::setMax_main_denominator(m_maxMainDenominator);
SettingsClass::writeConfig();
}void MainQtWidget::setupActions()
{ // new task action
m_NewTaskAction = new KAction(i18n("&새 연습"), "filenew",
KStdAccel::shortcut(KStdAccel::New),
this, SLOT(NewTask()),
actionCollection(), "NewTask");
// quit action
KStdAction::quit(kapp, SLOT(quit()), actionCollection());
//
KStdAction::preferences(this, SLOT( slotPrefs() ), actionCollection());
// a label just describing the Number of terms ComboBox
m_NrOfTermsLabel = new QLabel(i18n("항:"), 0, "kde toolbar widget");
m_NrOfTermsLabelAction = new KWidgetAction(m_NrOfTermsLabel, i18n("항:"),
ALT+Key_E,

```

```

    this, SLOT(NrOfTermsBoxSlot()),
    actionCollection(), "NrOfTermsLabelAction");
// the ComboBox holding possible values for term number
m_NrOfTermsBox = new QComboBox();
m_NrOfTermsBox->insertItem("2");
m_NrOfTermsBox->insertItem("3");
m_NrOfTermsBox->insertItem("4");
m_NrOfTermsBox->insertItem("5");
m_NrOfTermsBox->setCurrentItem(m_nrRatios - 2);
QToolTip::add( m_NrOfTermsBox, i18n( "원하는 항의 개수" ) );
QWhatsThis::add( m_NrOfTermsBox, i18n( "분수 계산시 원하는 항의 수를 선택하세요(2, 3, 4
or 5)" ) );
m_NrOfTermsBoxAction = new KWidgetAction(m_NrOfTermsBox, i18n("항의 수"),
ALT+Key_E, this, SLOT(NrOfTermsBoxSlot()), actionCollection(), "NrOfTermsBoxAction");
// now connect the ComboBox's signal textChanged() to the slot function
QObject::connect(m_NrOfTermsBox, SIGNAL(activated(int)), this,
SLOT(NrOfTermsBoxSlot()));
// a label just describing the max. main denominator ComboBox
m_MaxMainDenominatorLabel = new QLabel(i18n("최대. 주 분모:"), 0, "kde toolbar
widget");
m_MaxMainDenominatorLabelAction = new KWidgetAction(m_MaxMainDenominatorLabel,
i18n("최대. 주 분모:"), ALT+Key_D,
this, SLOT(MaxMainDenominatorBoxSlot()),
actionCollection(), "MaxMainDenominatorLabelAction");
// the ComboBox holding possible values for the max. main denominator
m_MaxMainDenominatorBox = new QComboBox(this);
m_MaxMainDenominatorBox->insertItem("10");
m_MaxMainDenominatorBox->insertItem("20");
m_MaxMainDenominatorBox->insertItem("30");
m_MaxMainDenominatorBox->insertItem("50");
QToolTip::add( m_MaxMainDenominatorBox, i18n( "분모가 가질 수 있는 최대 수" ) );
QWhatsThis::add( m_MaxMainDenominatorBox, i18n( "분모가 가질 수 있는 최대 수를
선택하세요: 10, 20, 30, 40 or 50." ) );
switch (m_maxMainDenominator)
{
case 10 : m_MaxMainDenominatorBox->setCurrentItem(0);
break;
case 20 : m_MaxMainDenominatorBox->setCurrentItem(1);
break;
case 30 : m_MaxMainDenominatorBox->setCurrentItem(2);
break;
case 50 : m_MaxMainDenominatorBox->setCurrentItem(3);

```

```

break;
}
m_MaxMainDenominatorBoxAction = new KWidgetAction(m_MaxMainDenominatorBox,
i18n("최대 분모"), ALT+Key_D, this, SLOT(MaxMainDenominatorBoxSlot()),
actionCollection(), "MaxMainDenominatorBoxAction");
// now connect the ComboBox's signal textChanged() to the slot function
QObject::connect(m_MaxMainDenominatorBox, SIGNAL(activated(int)),
this, SLOT(MaxMainDenominatorBoxSlot()));
// a label just describing the operation ComboBox
m_OperationLabel = new QLabel(i18n("연산들:"), 0, "kde toolbar widget");
m_OperationLabelAction = new KWidgetAction(m_OperationLabel, i18n("연산들:"),
ALT+Key_O,
this, SLOT(OperationBoxSlot()),
actionCollection(), "OperationLabelAction");
// the ComboBox holding possible combinations for operations
m_OperationBox = new QComboBox(this);
m_OperationBox->insertItem(i18n("덧셈/뺄셈"));
m_OperationBox->insertItem(i18n("곱셈/나눗셈"));
m_OperationBox->insertItem(i18n("전체 연산들 혼합"));
if (m_addSub == true && m_mulDiv == false)
{
m_OperationBox->setCurrentItem(0);
} else if (m_addSub == false && m_mulDiv == true) {
m_OperationBox->setCurrentItem(1);
} else if (m_addSub == true && m_mulDiv == true) {
m_OperationBox->setCurrentItem(2);
}
QToolTip::add( m_OperationBox, i18n( "원하신 연산들" ) );
QWhatsThis::add( m_OperationBox, i18n( "함수 계산 연습에서 사용할 연산들의 타입을
선택하세요: 덧셈/뺄셈, 곱셈/나눗셈, 전체 연산들 혼합. 전체 연산들 혼합 옵션을 선택할 경우,
덧셈, 뺄셈, 곱셈, 나눗셈중 하나가 임의로 선택됩니다." ) );
m_OperationBoxAction = new KWidgetAction(m_OperationBox, i18n("연산들:"), ALT+Key_O,
this, SLOT(OperationBoxSlot()), actionCollection(), "OperationBoxAction");
// now connect the ComboBox's signal textChanged() to the slot function
QObject::connect(m_OperationBox, SIGNAL(activated(int)), this, SLOT(OperationBoxSlot()));
#if (KDE_VERSION_MINOR>=3) && (KDE_VERSION_MAJOR>=3)
if (!initialGeometrySet())
resize( QSize(725, 330).expandedTo(minimumSizeHint()));
setupGUI(ToolBar | Keys | StatusBar | Create);
setAutoSaveSettings();
#endif
}/* ----- private slots ----- */

```

```
void MainQWidget::NewTask()
{
#ifdef DEBUG
    kdDebug() << "NewTask MainQWidget" << endl;
    kdDebug() << "pageIndex(m_taskview): " << m_exercises->pageIndex(m_taskview) << endl;
    kdDebug() << "pageIndex(m_exerciseCompare): " <<
        m_exercises->pageIndex(m_exerciseCompare) << endl;
    kdDebug() << "pageIndex(m_exerciseConvert): " <<
        m_exercises->pageIndex(m_exerciseConvert) << endl;
#endif
    // check which page should generate a new task
    switch (m_exercises->activePageIndex())
    {
    case 0 :
        m_taskview->forceNewTask();
        break;
    case 1 :
        m_exerciseCompare->forceNewTask();
        break;
    case 2 :
        m_exerciseConvert->forceNewTask();
        break;
    case 3 :
        m_exerciseFactorize->forceNewTask();
        break;
    }
    /* this doesn't seem to work, because pageIndex always returns 0
    if (m_exercises->activePageIndex() == m_exercises->pageIndex(m_taskview))
    {
        m_taskview->forceNewTask();
        return;
    }
    if (m_exercises->activePageIndex() == m_exercises->pageIndex(m_exerciseCompare))
    {
        m_exerciseCompare->forceNewTask();
        return;
    }
    */
    /* this even do not compile, but I don't know why
    switch (m_exercises->activePageIndex())
    {
    case m_exercises->pageIndex(m_taskview):
        break;
```

```
case m_exercises->pageIndex(m_exerciseCompare):
    m_exerciseCompare->forceNewTask();
    break;
}
*/
return;
}void MainQWidget::NrOfTermsBoxSlot()
{
#ifdef DEBUG
    kdDebug() << "MainQWidget::NrOfTermsBoxSlot()" << endl;
#endif
    QString curr_nr = m_NrOfTermsBox->currentText();
    m_MaxMainDenominatorBox->clear();
    if (m_mulDiv == true)
    {
        if (curr_nr == "2")
        {
            m_MaxMainDenominatorBox->insertItem("10");
            m_MaxMainDenominatorBox->insertItem("20");
            m_MaxMainDenominatorBox->insertItem("30");
            m_MaxMainDenominatorBox->insertItem("50");
            m_nrRatios = 2;
            m_maxMainDenominator = 10;
        } else if (curr_nr == "3") {
            m_MaxMainDenominatorBox->insertItem("20");
            m_MaxMainDenominatorBox->insertItem("30");
            m_MaxMainDenominatorBox->insertItem("50");
            m_nrRatios = 3;
            m_maxMainDenominator = 20;
        } else if (curr_nr == "4") {
            m_MaxMainDenominatorBox->insertItem("20");
            m_MaxMainDenominatorBox->insertItem("30");
            m_MaxMainDenominatorBox->insertItem("50");
            m_nrRatios = 4;
            m_maxMainDenominator = 20;
        } else {
            m_MaxMainDenominatorBox->insertItem("50");
            m_nrRatios = 5;
            m_maxMainDenominator = 50;
        }
        m_MaxMainDenominatorBox->setCurrentItem(0);
    } else {
        /* no multiplication or division allowed, so we add the default values */
```



```

m_MaxMainDenominatorBox->insertItem("10");
m_MaxMainDenominatorBox->insertItem("20");
m_MaxMainDenominatorBox->insertItem("30");
m_MaxMainDenominatorBox->insertItem("50");
if (curr_nr == "2")
    m_nrRatios = 2;
else if (curr_nr == "3")
    m_nrRatios = 3;
else if (curr_nr == "4")
    m_nrRatios = 4;
else
    m_nrRatios = 5;
} // if (m_mulDiv == true)
// set the new task parameters
(void) m_taskview->setTaskParameters(m_addSub, m_mulDiv, m_nrRatios,
m_maxMainDenominator);
}void MainQtWidget::MaxMainDenominatorBoxSlot()
#ifdef DEBUG
kdDebug() << "MainQtWidget::MaxMainDenominatorBoxSlot()" << endl;
#endif
// get the max. size from the ComboBox, convert it to a number and store
// it in the private member
QString curr_md = m_MaxMainDenominatorBox->currentText();
m_maxMainDenominator = curr_md.toInt();
// set the new task parameters
(void) m_taskview->setTaskParameters(m_addSub, m_mulDiv, m_nrRatios,
m_maxMainDenominator);
}void MainQtWidget::OperationBoxSlot()
#ifdef DEBUG
kdDebug() << "MainQtWidget::OperationBoxSlot()" << endl;
#endif
int index = m_OperationBox->currentItem(); // get selected item
// user has selected the operations for the next task, so store it in the
// private members
if (index == 0)
{
    m_addSub = true;
    m_mulDiv = false;
    /* set the number of terms box and max main denominator box correctly */
    NrOfTermsBoxSlot();
} else if (index == 1) {
    m_addSub = false;

```

```

m_mulDiv = true;
/* set the number of terms box and max main denominator box correctly */
NrOfTermsBoxSlot();
} else {
    m_addSub = true;
    m_mulDiv = true;
    /* set the number of terms box and max main denominator box correctly */
    NrOfTermsBoxSlot();
}
// set the new task parameters
(void) m_taskview->setTaskParameters(m_addSub, m_mulDiv, m_nrRatios,
m_maxMainDenominator);
}void MainQtWidget::slotPrefs()
{ // do not show dialog twice
if (KConfigDialog::showDialog("settings"))
return;
//KConfigDialog didn't find an instance of this dialog, so lets create it :
KConfigDialog* configDialog = new KConfigDialog( this, "settings", SettingsClass::self() );
TaskViewerOptionsBase * taskViewerOptions = new TaskViewerOptionsBase(0,
"TaskViewerOptionsBase");
    configDialog->addPage(taskViewerOptions, i18n("작업 보기 설정"), "colorize");
    // User edited the configuration - update your local copies of the
// configuration data
connect(configDialog, SIGNAL(settingsChanged()), this, SLOT(slotApplySettings()));
configDialog->show();
/*
SettingsDialog * dlg = new SettingsDialog(this);
connect(dlg, SIGNAL(configChanged()), this, SLOT(slotApplySettings()));
dlg->exec();
delete dlg;
dlg = NULL;
*/
return;
}void MainQtWidget::slotApplySettings()
{ // update the task view
m_taskview->update();
m_exerciseCompare->update();
m_exerciseConvert->update();
m_exerciseFactorize->update();
return;
}void MainQtWidget::slotAboutToShowPage(QWidget * page)
#ifdef DEBUG

```

```

kdDebug() << "slotAboutToShowPage MainQtWidget" << endl;
kdDebug() << "pageIndex(m_taskview): " << m_exercises->pageIndex(m_taskview) << endl;
kdDebug() << "pageIndex(m_exerciseCompare): " <<
m_exercises->pageIndex(m_exerciseCompare) << endl;
kdDebug() << "pageIndex(m_exerciseConvert): " <<
m_exercises->pageIndex(m_exerciseConvert) << endl;
#endif
// check which page to show
if (m_exercises->pageIndex(page) == m_exercises->pageIndex(m_taskview))
{
// exercise solve task with fraction (taskview.h)
m_NrOfTermsBox->setEnabled(true);
m_MaxMainDenominatorBox->setEnabled(true);
m_OperationBox->setEnabled(true);
} else {
m_NrOfTermsBox->setEnabled(false);
m_MaxMainDenominatorBox->setEnabled(false);
m_OperationBox->setEnabled(false);
}
return;
}bool MainQtWidget::queryExit()
{writeOptions();
return true;
}
#include "mainqtwidget.moc"

```

✱ taskvieweroptionsbase.ui (작업 옵션 설정 관련 UI 파일)

```

<!DOCTYPE UI><UI version="3.3" stdsetdef="1">
<class>TaskViewerOptionsBase</class>
<widget class="QWidget">
<property name="name">
<cstring>TaskViewerOptionsBase</cstring>
</property>
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>469</width>
<height>599</height>
</rect>
</property>

```

```

<property name="caption">
<string>TaskViewerOptionsBase</string>
</property>
<grid>
<property name="name">
<cstring>unnamed</cstring>
</property>
<widget class="QWidget" row="0" column="0">
<property name="name">
<cstring>layout2</cstring>
</property>
<vbox>
<property name="name">
<cstring>unnamed</cstring>
</property>
<widget class="KFontChooser">
<property name="name">
<cstring>kcfcg_taskFont</cstring>
</property>
<property name="sizePolicy">
<sizepolicy>
<hsizetype>1</hsizetype>
<vsizetype>3</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="toolTip" stdset="0">
<string>폰트의 크기 변경</string>
</property>
</widget>
<widget class="QGroupBox">
<property name="name">
<cstring>colorGroupBox</cstring>
</property>
<property name="sizePolicy">
<sizepolicy>
<hsizetype>5</hsizetype>
<vsizetype>0</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>

```

```

</property>
<property name="title">
<string>색상</string>
</property>
<widget class="QWidget">
<property name="name">
<cstring>layout7</cstring>
</property>
<property name="geometry">
<rect>
<x>10</x>
<y>20</y>
<width>230</width>
<height>110</height>
</rect>
</property>
<grid>
<property name="name">
<cstring>unnamed</cstring>
</property>
<widget class="KColorButton" row="1" column="1">
<property name="name">
<cstring>kcfg_operationColor</cstring>
</property>
<property name="sizePolicy">
<sizepolicy>
<hsizetype>1</hsizetype>
<vsizetype>0</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="text">
<string></string>
</property>
<property name="color">
<color>
<red>0</red>
<green>0</green>
<blue>255</blue>
</color>
</property>

```

```

<property name="toolTip" stdset="0">
<string>연산 기호의 색상 변경</string>
</property>
</widget>
<widget class="KColorButton" row="2" column="1">
<property name="name">
<cstring>kcfg_fractionBarColor</cstring>
</property>
<property name="text">
<string></string>
</property>
<property name="color">
<color>
<red>255</red>
<green>0</green>
<blue>0</blue>
</color>
</property>
<property name="toolTip" stdset="0">
<string>분수선의 색상 변경</string>
</property>
</widget>
<widget class="QLabel" row="2" column="0">
<property name="name">
<cstring>fractionBarLabel</cstring>
</property>
<property name="sizePolicy">
<sizepolicy>
<hsizetype>1</hsizetype>
<vsizetype>5</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="text">
<string>분수 선:</string>
</property>
</widget>
<widget class="QLabel" row="1" column="0">
<property name="name">
<cstring>operationLabel</cstring>
</property>

```

```

<property name="sizePolicy">
<sizepolicy>
<hsizetype>1</hsizetype>
<vsizetype>5</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>
<property name="text">
<string>연산 기호:</string>
</property>
</widget>
<widget class="KColorButton" row="0" column="1">
<property name="name">
<cstring>kcfg_numberColor</cstring>
</property>
<property name="text">
<string></string>
</property>
<property name="color">
<color>
<red>0</red>
<green>255</green>
<blue>0</blue>
</color>
</property>
<property name="toolTip" stdset="0">
<string>숫자의 색상 변경</string>
</property>
</widget>
<widget class="QLabel" row="0" column="0">
<property name="name">
<cstring>numberLabel</cstring>
</property>
<property name="sizePolicy">
<sizepolicy>
<hsizetype>1</hsizetype>
<vsizetype>5</vsizetype>
<horstretch>0</horstretch>
<verstretch>0</verstretch>
</sizepolicy>
</property>

```

```

<property name="text">
<string>숫자:</string>
</property>
</widget>
</grid>
</widget>
</widget>
<widget class="QGroupBox">
<property name="name">
<cstring>groupBox2</cstring>
</property>
<property name="title">
<string>일반</string>
</property>
<grid>
<property name="name">
<cstring>unnamed</cstring>
</property>
<widget class="QCheckBox" row="0" column="0">
<property name="name">
<cstring>kcfg_showSpecialRatioNotation</cstring>
</property>
<property name="text">
<string>결과를 1 2/3처럼 혼합된 숫자로 보여줍니다.</string>
</property>
<property name="toolTip" stdset="0">
<string>혼합된 숫자 기수법의 결과를 보여주는 여부를 선택할 수 있습니다.</string>
</property>
</widget>
</grid>
</widget>
</vbox>
</widget>
</grid>
</widget>
<layoutdefaults spacing="6" margin="11"/>
<includehints>
<includehint>kfontdialog.h</includehint>
<includehint>kcolorbutton.h</includehint>
<includehint>kcolorbutton.h</includehint>
<includehint>kcolorbutton.h</includehint>
</includehints>

```

</UI>

* 이 파일은 'Qt3 Designer' 프로그램을 통해 UI를 직접 변경 하였다.

3. KBruch 메뉴얼

목차

1장 소개글

2장 Kbruch 사용법

- 2.1 기본화면
- 2.2 통계적인 부분
- 2.3 화면 설정

3장 연습

- 3.1 분수 연습
 - 3.1.1 변수
 - 3.1.2 문제 풀기
- 3.2 비교 연습
- 3.3 전환 연습
- 3.4 인수분해 연습

4장 만든사람 및 License

부록 A 설치방법

- A.1 Kbruch 얻는 방법
- A.2 필요사항
- A.3 편집 및 설치

1장. 소개글

KBruch는 분수 계산 연습을 위한 작은 프로그램이다. 그래서 4개의 다른 연습들을 제공한다.

◎ 분수 연습 - 주어진 분수 문제를 푸는 연습이다. 분모와 분자를 입력하면 된다. 대표적인 연습이다.

◎ 비교 - 두개의 주어진 분수 크기를 비교하는 연습이다.

◎ 전환 - 주어진 숫자를 분수로 전환하는 연습이다.

◎ 인수분해 - 소수 인자를 이용해 주어진 숫자를 인수분해 하는 연습이다.

모든 KBruch의 연습들은 문제를 만들고 사용자는 그것을 묻다. 이 프로그램은 답을 체크하여 답에 대한 결과를 보여준다.

Kbruch는 얼마나 문제를 풀었는지, 그리고 얼마나 풀 문제가 맞았는지를 계산한다. 각 사용자에게 대해 통계를 보여준다. 하지만, 기본 화면의 통계를 보여주는 부분은 숨길 수 있다. 사용자들은 통계점수를 초기화 할 수 있다.

KBruch는 문제를 만드는 핵심이 매우 간결하고 뚜렷하다. 온라인 헬프 시스템은 서로 다른 상태에 대해 적절한 도움을 제공한다.

KBruch는 Free이고 라이선스는 GNU Public License를 따른다.

2장. Kbruch 사용법

2.1 기본화면

다음은 KBruch의 기본 화면 스크린샷이다:



<그림 3> KBruch Main 화면

모든 행동은 이 화면에서 진행된다. KBruch는 나이 어린 사용자가 사용하기에 매우 쉽다. 이 화면은 5부분으로 분리된다.

- 메뉴바는 작업, Setting, Help 3개로 구성되어 있다.
- 툴바는 주어지는 분수 문제의 난이도를 조절할 수 있다.
- 연습을 선택하는 곳에서는 다른 연습문제로 전환 할 수 있다.
- 문제 출제가 되는 부분은 주어진 문제의 결과가 출력된다.
- 통계 부분은 문제를 얼마나 많이 풀었는지와 몇개 맞았는지 보여준다.

먼저, 화면의 왼쪽부분 연습을 선택하는 곳에서 원하는 연습을 선택한다. 문제 출제가 되는 부분이 선택된 연습에 따라 바뀐다. 만약 분수 연습을 선택하면 툴바의 settings 부분이 활성화 될 것이다.

툴바안에서 '새 작업' 아이콘을 찾을 수 있다. 이 아이콘은 메뉴의 '작업 -> 새작업' 과 같다. 이 작업은 항상 활성화 되어 있다. 만약 현재 주어진 문제를 풀지 못하면, 새 작업이 만들어지고, 틀린 답으로 계산된다.

통계가 나오는 화면은 세로줄을 오른쪽으로 드래그 함으로써 숨길 수 있다.

만약 현재 선택해서 연습하고있는 Kbruch를 끝내면 저장되고 다음 시작시 복구된다.

2.2 통계 부분

이 부분은 기본 화면에서 볼 수 있다:

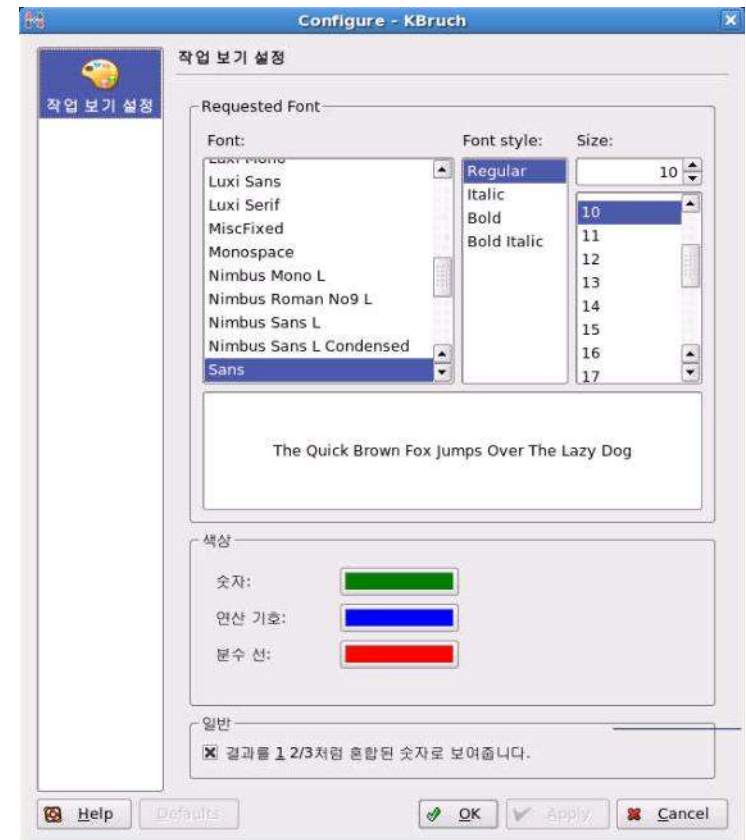
- 얼마나 문제를 풀었는가
- 몇개의 문제를 맞추었는가
- 몇개의 문제를 틀렸는가

숫자 아래 '재설정' 버튼을 누르면 통계가 보이는 부분은 재설정된다.

통계부분은 KBruch를 끝내면 저장되고 다음 시작시 복구된다.

2.3 화면 설정

기본화면의 화면 설정 부분은 문제 출제에 대한 일반적인 설정을 조정한다. 이 대화상자는 Settings -> Configure KBruch를 열면 된다.



<그림 4> 화면 설정

대화상자의 가장 윗부분에는 폰트를 선택할 수 있게 되어 있다. 아래부분에는 수리논리학 표현의 색상을 선택할 수 있게 되어 있다. 혼합된 기수법에 대한 결과를 볼 수도 있다. 이 기수법은 어디에서나 사용되지는 않는다. 그래서 비활성화 할 수 있다. 설정은 KBruch를 끝내면 저장되고 다음 시작시 복구된다.

3장. 연습

3.1 분수 연습

이 연습은 주어진 문제를 푸는 것이다. 그때문에 분모와 분자를 입력한다. 일반적인 문제의 난이도는 매개 변수를 통해 결정할 수 있다.

3.1.1 변수

문제의 난이도는 3개의 변수에 영향을 받는다.

항 문제에 항의 수가 주어진다. 2부터 5까지 포함된다.

최대 주 분모 분포의 최대 수를 지정한다. 최소 10부터 최대 50까지 지정한다.

연산자 연산자들을 문제에 사용할 수 있다. 덧셈/뺄셈, 곱셈/나눗셈 또는 모든 연산을 사용한다.

변수를 변경한 후에 톨바의 '새작업' 버튼을 누르면 새 변수를 사용해서 문제가 만들어진다. 이 작업은 메뉴바의 '작업 -> 새작업'을 사용해도 동일하다.

노트

최대 주 분모는 항의 수를 선택 하면 기본 값으로 재설정된다. 그래서 먼저 항의 수를 선택 한 후 최대 주 분모를 선택한다.

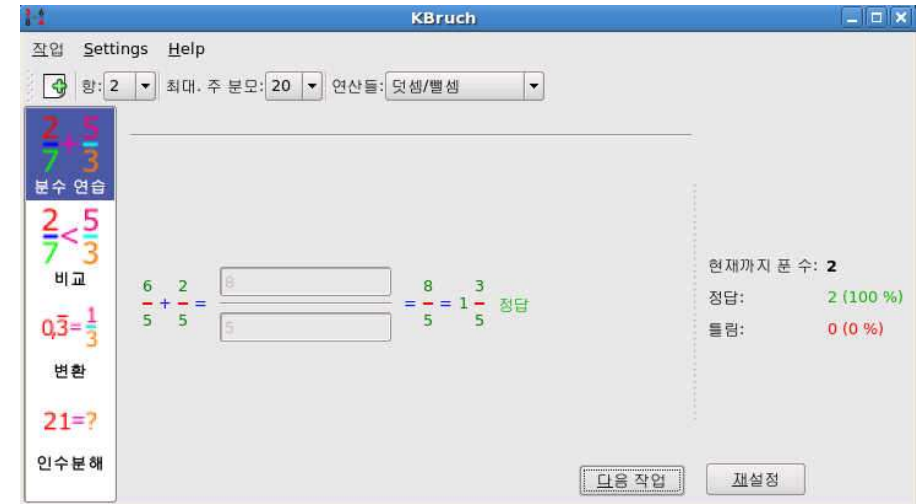
만약 현재 문제를 풀지 않고 '새 작업'을 부르게 되면, 그 문제는 틀린 문제로 계산된다. 이것을 방지하기 위해 문제를 풀고 새로운 문제를 받기 전에 변수를 변경해야 한다. 변수 선택은 KBruch를 끝내면 저장되고 다음 시작시 복구된다.

3.1.2 문제 풀기

주어진 문제를 풀기 위해, 두개의 입력 상자에 결과를 입력해야 한다. 윗 상자에는 분자를 아래 상자에는 분모를 넣는다.

만약 결과가 음수이면, 분모나 분자의 앞에 빼기 기호를 넣을 수 있다. 만일 결과가 0이면, 분자 입력 상자에 0을 넣는다. 만일 결과가 1의 분모값을 가지면 아래 상자를 비워놓을 수 있다. 결과를 입력한 후에 입력상자 아래 '작업 체크'버튼을 누른다. Kbruch는 입력값을

검색해서 올바른 결과를 보여준다. 따라서 화면은 작게 변화하며, 다음 스크린 샷처럼 보여준다.

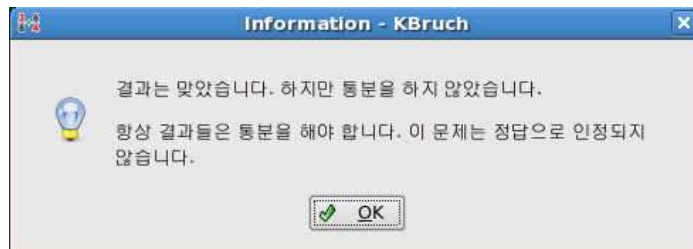


<그림 5> 분수 연습 화면

이 작업은 정답을 맞추었다. 정답은 두개의 다른 폼을 보여준다. '정답' 앞에있는 두번째 폼은 몇몇 나라에서 사용하는 폼이고 1+ 3/5 라고 읽어야 한다.

노트

항상 결과를 통분해야 한다. 왜냐하면 대부분의 선생님들이 통분한 결과에만 점수를 주기 때문이다. KBruch는 만약 맞는 결과일 지라도 통분을 하지 않았을 경우에는 다음 스크린샷과 같은 짧은 메시지를 출력해 준다.



<그림 6> 에러 메시지

다음 문제를 계속하려면, '다음 작업' 버튼을 누르면 된다. 만약 다음 문제부터 변수를 바꾸기 원한다면, '다음 작업' 버튼을 누르기 전에 변경해야 한다.

3.2 비교 연습

이 연습은 두개의 주어진 분수를 비교하는 문제이다. 둘중 더 큰 분수를 선택한 후 맞는 비교 기호를 결정한다.

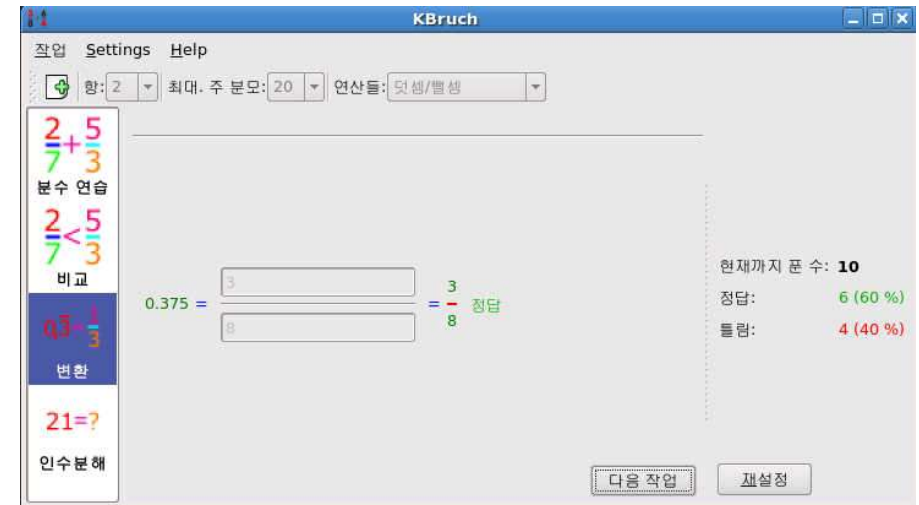


<그림 7> 비교연습 화면

먼저 맞는 비교 기호를 선택한다. 비교 기호 버튼을 클릭하여 변화 시킬 수 있다. 비교 기호 버튼을 클릭하여 선택 한 후에 '작업 체크' 버튼을 누른다. KBruch는 입력값을 검색하여 올바른 결과를 출력해준다. 이 후에 '다음 작업' 버튼을 눌러 다음 문제를 풀 수 있다.

3.3 전환 연습

이 연습은 주어진 숫자를 분수로 전환하는 문제이다. 분모와 분자에 숫자를 넣으면 된다.



<그림 8> 변환연습 화면

분모와 분자에 값을 넣은 후 '작업 체크' 버튼을 누른다. KBruch는 입력값을 검색해서 올바른 결과를 출력해 준다. 이 후에 '다음 작업' 버튼을 눌러 다음 문제를 풀 수 있다.

노트

결과를 통분하는 것을 잊으면 안된다. 통분하지 않은 결과는 정답으로 인정되지 않는다.

3.4 인수분해 연습

이 연습은 소수를 이용해서 주어진 수를 인수분해 하는 것이다. 모든 소수를 가지고 연습한다.



<그림 9> 인수분해 연습 화면

소수 숫자를 눌러 인수분해 한다. 현재 입력한 소수는 인수분해 한 수로 화면에 나타난다. 만일 마지막에 입력한 소수를 지우고 싶다면, '마지막 인자를 지움' 버튼을 누르면 된다. 모든 소수를 입력한 후에 '작업 체크' 버튼을 누른다. KBruch는 입력값을 검사하여 올바른 값을 출력한다. 이 후에 '다음 작업' 버튼을 눌러 다음 문제를 풀 수 있다. 스크린 샷의 가장 꼭대기에 소수 사이의 '*' 기호를 볼 수 있다. 이 기호는 곱셈 기호이다.

노트

모든 소수를 입력할 때, 소수를 여러번 반복 입력해도 된다.

예제 : 18의 소수는 2, 3, 3이다.

4장. 만든 사람 및 License

Kbruch

Program and documentation Copyright, 2001-2004 Sebastian Stein

seb.kde@hpfs.de

Thank you goes to:

- Anne-Marie Mahfouf for managing the KDE-edu project
- Klas Kalass for helping design the interface

- Eva Brucherseifer for writing Kmath(KBruch's GUI is based on this work)
- Dominique Devriese for bug fixing
- Stefan Schumacher for testing a very early version
- Sven Guckes for testing the text interface (which has been removed)
- David Faure for helping with KDE CVS and KDE Bugzilla
- Robert Gogolok for helping with KDE CVS
- John Kesson for proof reading the handbook

This documentation is licensed under the terms of the GNU Free Documentation License.

This program is licensed under the terms of the GNU General Public License.

부록 A. 설치 방법

A.1 Kbruch얻는 방법

KBruch는 KDE 프로젝트의 한 부분입니다. <http://www.kde.org/>

KBruch는 kdeedu 패키지 안에서 찾을 수 있습니다.

KDE프로젝트의 메인 ftp 사이트 입니다. <ftp://ftp.kde.org/pub/kde>

A.2 필요사항

KBruch를 성공적으로 실행하기 위해서는 KDE 3.1이상의 버전이 필요합니다.

KBruch는 Kbruch 홈페이지나 KDE-edu 프로젝트에서 찾을 수 있습니다.

CVS 메시지를 통해 변환된 리스트를 찾을 수 있습니다.

A.3 편집 및 설치

KBruch를 시스템에 컴파일 설치 하려면 KBruch의 기본 디렉토리에서 다음 사항을 입력해야 합니다.

```
% ./configure
% make
% make install
```

KBruch를 사용하는데 autoconf나 automake를 사용하여 컴파일해도 문제 없습니다. 문제가 발생하는 경우 KDE 메일링 리스트에 보고해 주시기 바랍니다.