目录

[第一章 程序简介 2](#_Toc186660771)

[1. 背景 2](#_Toc186660772)

[2. 简介 2](#_Toc186660773)

[3. 目标用户 2](#_Toc186660774)

[4. 主要功能 2](#_Toc186660775)

[5. 优势与创新 2](#_Toc186660776)

[第二章 程序运行环境 4](#_Toc186660777)

[1. 程序硬件环境 4](#_Toc186660778)

[2. 系统软件环境 4](#_Toc186660779)

[第三章 程序主要功能 5](#_Toc186660780)

[1. 代码编辑 5](#_Toc186660781)

[2. 代码运行 6](#_Toc186660782)

[3. 代码编译 6](#_Toc186660783)

[第四章 Brainfxxk语言简介 8](#_Toc186660784)

[1. 简介 8](#_Toc186660785)

[2. 字符标识 8](#_Toc186660786)

[3. 内存模型 9](#_Toc186660787)

[4. 示例代码 9](#_Toc186660788)

# 

# 程序简介

## 背景

Brainfxxk 是一种极简主义的编程语言，以其简洁却极具挑战性的语法而闻名。为了满足编程爱好者、教育工作者以及开发者对 Brainfxxk 的学习、使用和应用需求，Brainfxxk 编译器应运而生。

这款工具不仅支持 Brainfxxk 程序的编写与调试，还能高效地将代码转换为可执行文件，为用户带来无缝的开发体验。

## 简介

Brainfxxk 编译器是一个轻量级的应用程序，专为编译和运行 Brainfxxk 代码设计。它提供了一个简单的编辑界面，支持输入 Brainfxxk 程序并直接运行，实时展示程序的输出结果。同时，用户还可以将代码编译为可执行文件（.exe），方便直接运行而无需依赖外部解释器。该工具旨在为用户提供一个简洁直观的开发环境，帮助学习和调试 Brainfxxk 程序。

## 目标用户

编程初学者：通过挑战性的语言学习基本编程思维。

计算机科学教育者：用于教学展示极简编程语言的核心概念。

开发爱好者：体验极简语言开发的乐趣和挑战。

编程竞赛选手：快速测试和验证 Brainfxxk 代码的执行效果。

## 主要功能

代码编辑器：支持语法高亮、代码检查与错误提示。

实时运行：直接运行 Brainfxxk 程序，展示输出结果和内存状态。

编译功能：将代码转换为独立的可执行文件，便于分发和运行。

友好界面：直观的操作面板，降低使用门槛。

## 优势与创新

极简的使用体验：专为 Brainfxxk 语言优化，无需额外配置，即装即用。

实时反馈：运行和调试过程中提供即时结果，助力高效开发。

独立编译功能：生成可执行文件，突破语言运行环境的限制。

教学利器：通过内存监控和可视化展示，帮助用户深入理解语言原理。

# 程序运行环境

## 程序硬件环境

搭载了Windows操作系统的计算机。

## 系统软件环境

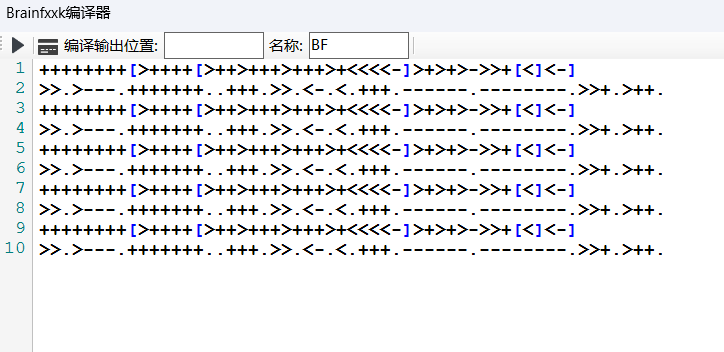
操作系统：Windows10及以上

运行平台：.Net Framework 4.7.2

# 程序主要功能

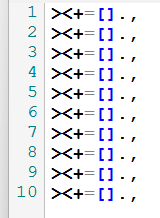
## 代码编辑

在编辑窗口中编写 Brainfxxk 代码。



支持语法高亮：

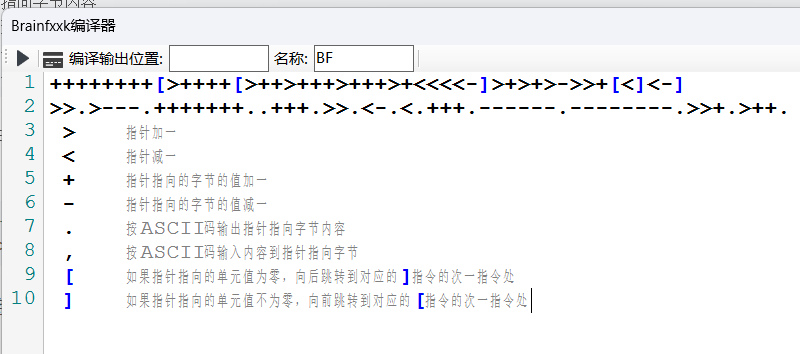
* > 和 <：指针操作。
* + 和 -：数据增减。
* [ 和 ]：循环指令。
* . 和 ,：输出和输入。



非 Brainfxxk 指令将以灰色显示，便于区分。

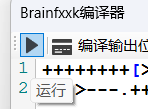
在 Brainfxxk 编程语言中，只有八个有效的指令字符：>, <, +, -, ., ,, [, 和 ]，这些是用于控制指针、数据、输入输出和循环的基本操作。

非 Brainfxxk 指令指的是在 Brainfxxk 代码中出现的其他字符，它们不属于 Brainfxxk 语言的有效指令。在大多数 Brainfxxk 编译器中，这些字符会被忽略，因为它们对程序的行为没有影响。常见的非 Brainfxxk 指令字符包括空格、换行符、字母、数字等。

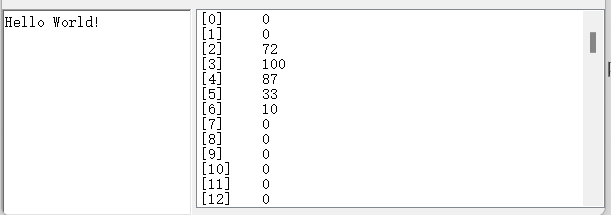


## 代码运行

点击 “运行” 按钮，可以直接运行当前的 Brainfxxk 程序。

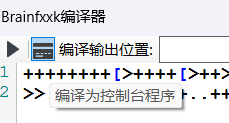


运行后，可以在结果窗口查看输出，同时在数据窗口查看内存数据变化。



## 代码编译

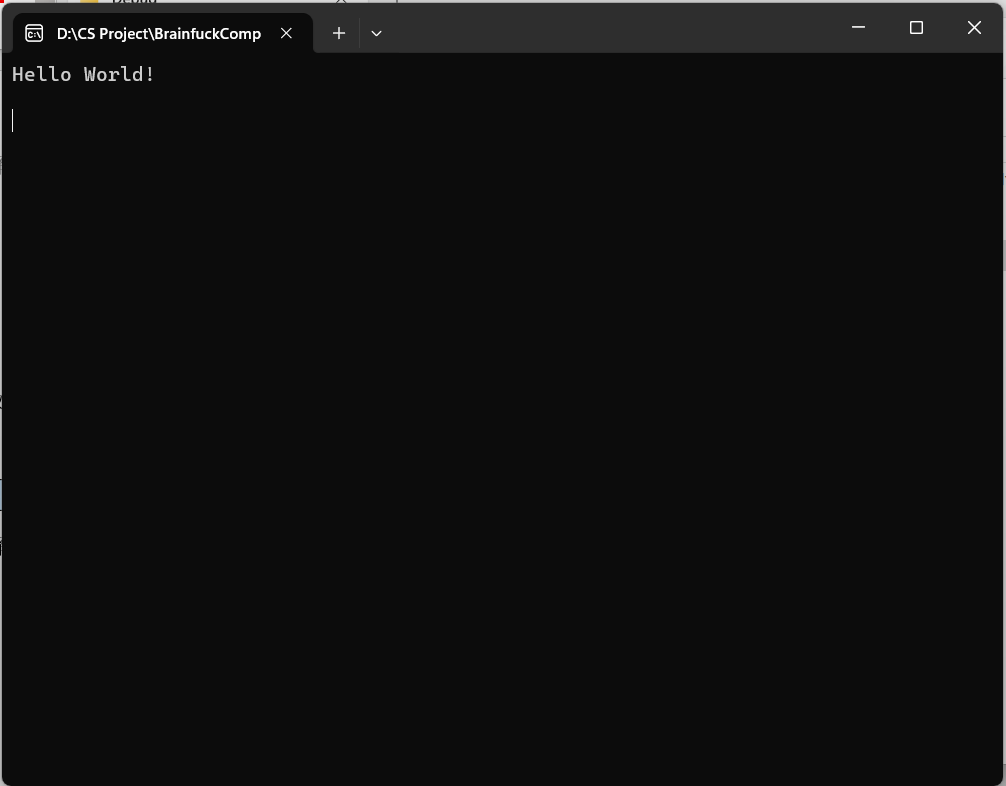
点击 “编译” 按钮，将 Brainfxxk 代码编译为独立的可执行文件。



可指定文件名和保存路径，编译完成后会自动打开目标文件所在的文件夹。



双击编译后的应用程序即可运行它。



# Brainfxxk语言简介

Brainfuck是一种极小化的计算机语言，它是由Urban Müller在1993年创建的。由于fuck在英语中是脏话，这种语言有时被称为brainf\*ck或brainf\*\*k，甚至被简称为BF。在此文档中，此语言被称为Brainfxxk。

## 简介

Müller的目标是建立一种简单的、可以用最小的编译器来实现的、符合图灵完全思想的编程语言。这种语言由八种状态构成，为Amiga机器编写的编译器（第二版）只有240个字节大小。

就象它的名字所暗示的，Brainfxxk程序很难读懂。尽管如此，Brainfxxk图灵机一样可以完成任何计算任务。虽然Brainfxxk的计算方式如此与众不同，但它确实能够正确运行。

这种语言基于一个简单的机器模型，除了指令，这个机器还包括：一个以字节为单位、被初始化为零的数组、一个指向该数组的指针（初始时指向数组的第一个字节）、以及用于输入输出的两个字节流。

这种语言，是一种按照“Turing complete（图灵完备）”思想设计的语言，它的主要设计思路是：用最小的概念实现一种“简单”的语言，Brainfxxk语言只有八种符号，所有的操作都由这八种符号的组合来完成。

## 字符标识

|  |  |
| --- | --- |
| 字符 | 含义 |
| > | 指针加一 |
| < | 指针减一 |
| + | 指针指向的字节的值加一 |
| - | 指针指向的字节的值减一 |
| . | 输出指针指向的单元内容（ASCII码） |
| , | 输入内容到指针指向的单元（ASCII码） |
| [ | 如果指针指向的单元值为零，向后跳转到对应的]指令的次一指令处 |
| ] | 如果指针指向的单元值不为零，向前跳转到对应的[指令的次一指令处 |

## 内存模型

Brainfxxk 使用一个线性数组（通常大小为 30,000 个内存单元）。每个内存单元初始值为零。指针始终指向当前正在操作的内存单元。

## 示例代码

++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]

>>.>---.+++++++..+++.>>.<-.<.+++.------.--------.>>+.>++.

这段代码通过一系列的内存操作和循环，最终输出“Hello World!”。