列举十种编译器：

1. **Microsoft Visual C++**

   支持语言：C 语言，C++，C++/CLI

   出现时间：1992

   代表人物：微软

   发展过程：第一代版本在 1992 年推出，可同时支援 16 位处理器与 32 位处理器版。经过多年发展完善，于 2002 年发行的 Visual C++ 7.0 集成了 MFC7.0，支持链接时代码生成和调试执行时检查。2013 年开发环境亦内置了源代码染色的功能。随后 MSVC 整合到了 visual studio 中。

   与其他编译器对比优势：以拥有“语法高亮”，IntelliSense（自动完成功能）以及高级除错功能而著称。比如，它允许用户进行远程调试，单步执行等。还有允许用户在调试期间重新编译被修改的代码，而不必重新启动正在调试的程序。其编译及建置系统以预编译头文件、最小重建功能及累加连结著称。这些特征明显缩短程式编辑、编译及连结花费的时间，在大型软件计划上尤其显著。

2. **Clang**

   支持语言：C 语言、C++、Objective-C

   出现时间：2007

   代表人物：由 Apple Google 开发

   发展过程：Clang 是一个由 Apple 主导编写，基于 LLVM 的 C/C++/Objective-C 编译器。2013 年 4 月,Clang 已经全面支持 C++11 标准，并开始实现 C++1y 特性。它与 GNU C 语言规范几乎完全兼容（当然，也有部分不兼容的内容，包括编译命令选项也会有点差异），并在此基础上增加了额外的语法特性，比如 C 函数重载（通过__attribute__((overloadable))来修饰函数），其目标（之一）就是超越 GCC。

   与其他编译器对比优势：支持其普通 lambda 表达式、返回类型的简化处理以及更好的处理 constexpr 关键字。终端用户特色：快速编译和较少内存占用[3]；有诊断功能；兼容 GCC。实用工具和应用：基础架构模块库；可满足多样的客户需求（代码重构，动态分析，代码生成，等）；允许集成到各种 IDE 中；使用 LLVM'BSD'协议；内部设计与实现：一个不可移植的代码库；一个为 C 语言、Objective-C、C++、Objective-C++设计的非通用(Non general-purpose)的语法解析器；与 C/C++/Objective-C 及它们的衍生语言具有高度一致性；

3. GCC

   支持语言：C、C++、Objective-C、Fortran、Java、Ada、Go 语言和 D 语言

   出现时间：1987

   代表人物：Richard Stallman

   发展过程：GCC 首个公开发布版本是在 1987 年由 Richard Stallman 发布的。它原本只能处理 C 语言。GCC 很快地扩展，变得可处理 C++。后来又扩展能够支持更多编程语言。GCC 4.7.0 带来了一组关于链接时优化 (LTO) 框架可提升伸缩性和降低内存使用，据开发者称，在 64 位系统上需要 8G 内存来对 Firefox 进行优化，然而用了 LTO 后只需 3G。

   与其他编译器对比优势：GCC 的最佳化技巧依其释出版本而有很大不同，但都包含了标准的最佳化算法，例如循环最佳化、执行绪跳跃、共通程序子句消减、指令排程等等。

4. G++

支持语言： C++
出现时间：1987
代表人物： Richard Stallman
发展过程：由 the Free Software Foundation 发行，与 gcc 共同组成 GNU 的编译器套件。发展历程大致同 gcc

与其他编译器对比：g++一般处理.cpp 文件，处理过程包含 gcc 的过程。过程是首先预处理，生成.i 的文件[预处理器 cpp]。然后将预处理后的文件转换成汇编语言，生成文件.s[编译器 egcs]。再由汇编变为目标代码（机器代码）生成.o 的文件[汇编器 as]。最后连接目标代码，生成可执行程序[链接器 ld]。

5. **IBM XL C++**

支持语言：C C++
出现时间：90 年代
代表人物：IBM
发展过程： 为 IBM 平台开发的 C/C++编译器，后来渐渐适用于多个操作系统。

与其他编译器对比优势：在 IBM 特定硬件与平台上表现非常好，Benchmark 性能测试非常优秀，其最初的设计思想就是为了性能，其最高优化级别可以达到 O5，带来的缺点则是在编译时间上往往过长。而另外一个常被人诟病的是对 C++标准的支持、开源软件的支持、错误信息的提示上都不够友好，然而从 13.2 开始，随着 IBM 采用 Clang 融合方案，目前这几项都得到有效改良。

6. **dev C++**

支持语言： C C++
出现时间：90 年代
代表人物： Bloodshed Orwell
发展过程：原开发公司 Bloodshed 在开发完 4.9.9.2 后停止开发，所以现在由 Orwell 公司继续更新开发，最新版本：5.11。

与其他编译器对比优势：Dev-C++是一个 Windows 环境下的一个适合于初学者使用的轻量级 C/C++ 集成开发环境（IDE），它是一款自由软件，遵守 GPL 许可协议分发源代码。它集合了 MinGW 中的 GCC 编译器、GDB 调试器和 AStyle 格式整理器等众多自由软件。Dev-C++ 的优点是功能简洁，适合于在教学中供 C/C++语言初学者使用。

7. **Microsoft Visual Studio**

支持语言：C C++ C# Python JAVA 等
出现时间：1997
代表人物： 微软
发展过程： 1997 年，微软发布了 Visual Studio 97。包含有面向 Windows 开发使用的 Visual Basic 5.0、Visual C++ 5.0，面向 Java 开发的 Visual J++和面向数据库开发的 Visual FoxPro，还包含有创建 DHTML (Dynamic HTML) 所需要的 Visual InterDev。2002 年开发了新语言 C#。自 2013 年起，每两年更新一次。

与其他编译器对比优势：组件众多，可以编译多种语言。功能复杂且全面。

8. **IDLE**

支持语言： Python

出现时间：1998

代表人物：Guido van Rossum

发展过程：与 Python 这门语言的发展过程相同，在 Python 被创造出来时由 Python 之父编写。在电脑中安装了 python 后，会自带 IDLE。

与其他编译器对比优势：Python 自带的集成开发环境，IDLE 具备语法高亮功能，还允许你在 IDLE 中运行你的程序。许多事情它会自动帮你处理。比如 debug。基本功能：语法加亮；段落缩进；基本文本编辑；TABLE 键控制；调试程序。Idle 总的来说是标准的 Python 发行版，甚至是由 Guido van Rossum 亲自编写（至少最初的绝大部分）。您可在能运行 Python 和 TK 的任何环境下运行 Idle。打开 Idle 后出现一个增强的交互命令行解释器窗口（具有比基本的交互命令提示符更好的剪切-粘贴、回行等功能）。除此之外，还有一个针对 Python 的编辑器（无代码合并，但有语法标签高亮和代码自动完成功能）、类浏览器和调试器。菜单为 TK “剥离”式；也就是点击顶部任意下拉菜单的虚线将会将该菜单提升到它自己的永久窗口中去。特别是 "Edit" 菜单，将其“靠”在桌面一角非常实用。Idle 的调试器提供断点、步进和变量监视功能；但并没有其内存地址和变量内容存数或进行同步和其他分析功能来得优秀。

9. **PyCharm**

支持语言：Python

出现时间：2010

代表人物：JetBrains

发展过程：2010 年发布第一版，随后 1.5 版做了大量 IDE 改进和 bug 修复。新版本对集成调试器、嵌入式本地终端等做了大量改进。

与其他编译器对比优势：PyCharm 是一种 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。PyCharm 拥有一般 IDE 具备的功能，比如， 调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。另外，PyCharm 还提供了一些很好的功能用于 Django 开发，同时支持 Google App Engine，更酷的是，PyCharm 支持 IronPython。

10. **Emacs**

支持语言： Python

出现时间：1976

代表人物： Richard Stallman Guy Steele

发展过程： Emacs 在 1970 年代诞生于 MIT 人工智能实验室（MIT AI Lab），最开始只作为一款文本编辑器使用。最初的 Emacs 同 TECO 一样只能运行在 PDP-10 系统。Emacs 虽然是在 TECO 的基础上发展起来的，不过它已经足以被认为是一个新的文本编辑器了。很快，Emacs 就成为 ITS 上的标准编辑程序，接着由 Michael McMahon 移植到 Tenex 和 TOPS-20 系统上。

与其他编译器对比优势：Emacs，著名的集成开发环境和文本编辑器。Emacs 被公认为是最受专业程序员喜爱的代码编辑器之一。Emacs 是目前世界上最具可移植性的重要软件之一，能够在当前大多数操作系统上运行，包括类 Unix 系统（GNU/Linux、各种 BSD、Solaris、AIX、IRIX 等等）、MS-DOS、Microsoft Window 以及 OpenVMS 等，还有

移动 Android 平台以及 iOS。Emacs 既可以在文本终端也可以在图形用户界面（GUI）环境下运行。在类 Unix 系统上，Emacs 使用 X Window 产生 GUI，或者直接使用“框架”（widget toolkit），例如 Motif、LessTif 或 GTK+等等。Emacs 也能够利用 Mac OS X 和 Microsoft Windows 的本地图形系统产生 GUI。用 GUI 环境下的 Emacs 能提供菜单（Menubar）、工具栏（toolbar）、scrollbar 以及 context menu 等交互方式。