

Q1

Interpretadores – Matlab, Ruby, PHP.

Q2

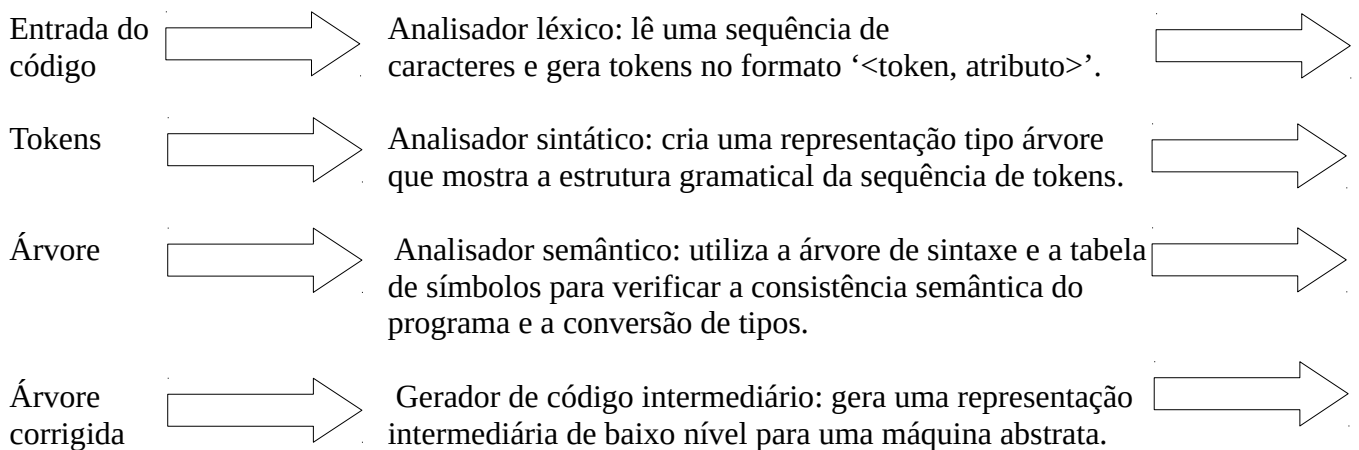
Fica mais fácil para depurar o compilador em caso de bugs e para saber o comportamento do código no pipeline e cache do processador.

Q3

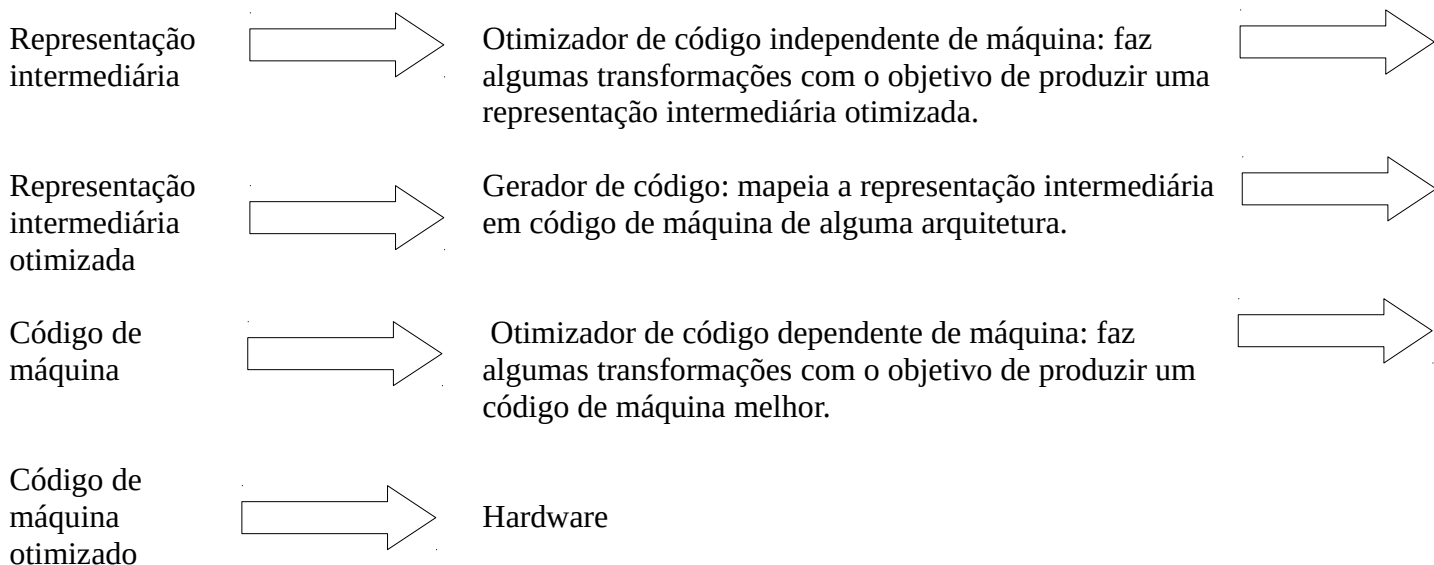
Eles conseguem armazenar valores flutuantes ‘float’.

Q4

Front-end



Back-end



Q5

- **GNU GCC**

- **Plataformas suportadas:**

Segundo o site Linfo: <http://www.linfo.org/gcc.html>

3b1	AMD 29k	AIX385	DEC Alpha
Altos3068	Amix	ARM	Convex
CRDS	Elxsi	FX2800	FX80
Genix	HP320	Clipper	Intel IA-64
Iris	i860	i960	Irix4
68000	Motorola m88k SvsV.3	MIPS-news	mot3300
NeXT	NS32K	NWS3250-v.4	HP-PA
PC532	Plexus	Pyramid	ROMP
RS/6000	SPARC-SunOs	SPARC-Solaris2	SPARC-SysV.4
SPUR	Sun386	Tahoe	TOW
Umpis	VAX		

E mais algumas plataformas da família x86

MS-DOS	ISC	SCO	SysV.3
SysV.4	Mach	BSD	Linux
Microsoft Windows	OS/2		

- **Linguagens suportadas:**

Segundo a Wikipédia: https://en.wikipedia.org/wiki/GNU_Compiler_Collection

GCC 10.2

C (gcc)	C++ (g++)	Objective-C	Fortran (gfortran)
Ada (GNAT)	Go (gccgo)	D (GDC, desde 9.1)	

GCC < 7

Linguagens acima e também Java (gcj).

GCC de terceiros

Pascal (gpc)	Modula-2	Modula-3	PL/I	VHDL (ghdl)
--------------	----------	----------	------	-------------

Algumas versões experimentais suportam GCC UPC compiler para Unified Parallel C ou Rust.

- **LLVM**

- **Plataformas suportadas:**

Segundos o site do LLVM: <https://llvm.org/docs/GettingStarted.html>

SO	Arch	Compiladores
Linux	x86	GCC, Clang
Linux	amd64	GCC, Clang
Linux	ARM	GCC, Clang
Linux	Mips	GCC, Clang
Linux	PowerPC	GCC, Clang
Linux	SystemZ	GCC, Clang
Solaris	V9 (Ultrasparc)	GCC
FreeBSD	x86	GCC, Clang
FreeBSD	amd64	GCC, Clang
NetBSD	x86	GCC, Clang
NetBSD	amd64	GCC, Clang
macOS	PowerPC	GCC
macOS	x86	GCC, Clang
Cygwin/Win32	x86	GCC
Windows	x86	Visual Studio
Windows x64	x86-64	Visual Studio

- **Linguagens suportadas:**

Segundo a Wikipédia: <https://en.wikipedia.org/wiki/LLVM>

C	C++	D	Delphi	Fortran
Haskell	Julia	Objective-C	Rust	Swift