т .		<i>,</i> .	$\alpha$
כו	borat	Orio	1 11 1
டு	ixiat	UI IU	$\cdots$

Interpretadores – Ma	tlab, Ruby, PHP.
<b>Q2</b> Fica mais fácil para codigo no pipeline e cache d	depurar o compilador em caso de bugs e para saber o comportamento do o processador.
Q3 Eles conseguem arma	azenar valores flutuantes 'float'.
<u>Q4</u>	
<u>Front-end</u>	
Entrada do código	Analisador léxico: lê uma sequência de caracteres e gera tokens no formato ' <token, atributo="">'.</token,>
Tokens	Analisador sintático: cria uma representação tipo árvore que mostra a estrutura gramatical da sequência de tokens.
Árvore	Analisador semântico: utiliza a árvore de sintaxe e a tabela de símbolos para verificar a consistência semântica do programa e a conversão de tipos.
Árvore corrigida	Gerador de código intermediário: gera uma representação intermediária de baixo nível para uma máquina abstrata.
Back-end	
Representação intermediária	Otimizador de código independente de máquina: faz algumas transformações com o objetivo de produzir uma representação intermediária otimizada.
Representação intermediária otimizada	Gerador de código: mapeia a representação intermediária em código de máquina de alguma arquitetura.
Código de máquina	Otimizador de código dependente de máquina: faz algumas transformações com o objetivo de produzir um código de máquina melhor.
Código de máquina otimizado	Hardware

### <u>Q5</u>

#### GNU GCC

### • Plataformas suportadas:

Segundo o site Linfo: <a href="http://www.linfo.org/gcc.html">http://www.linfo.org/gcc.html</a>

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	

# Microsoft Windows OS/2

• Linguagens suportadas:

Mach

Segundo a Wikipédia: <a href="https://en.wikipedia.org/wiki/GNU\_Compiler\_Collection">https://en.wikipedia.org/wiki/GNU\_Compiler\_Collection</a>

#### GCC 10.2

SysV.4

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

BSD

Linux

#### <u>GCC < 7</u>

Linguagens acima e também Java (gcj).

### GCC de terceiros

Pascal (gpc)	Modula-2	Modula-3	PL/I	VHDL (ghdl)
,01				, ,

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

### • <u>LLVM</u>

## • Plataformas suportadas:

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

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: <a href="https://en.wikipedia.org/wiki/LLVM">https://en.wikipedia.org/wiki/LLVM</a>

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