

1. B

解析1:

本题考查python语言的语法相关。

其中支持Python语言深度学习的有:

TensorFlow: Tensorflow拥有多层级结构, 可部署于各类服务器、PC终端和网页并支持GPU和TPU高性能数值计算, 被广泛应用于谷歌内部的产品开发和各领域的科学研究, 支持Python语言深度学习。

PyTorch: PyTorch是一个针对深度学习, 并且使用GPU和CPU来优化的tensor library (张量库) 是由Torch7团队开发, 是一个以Python优先的深度学习框架, 不仅能实现强大的GPU加速, 同时还支持动态的神经网络。

Keras: Keras是一个由Python编写的开源人工神经网络库, 可以作为Tensorflow、Microsoft-CNTK和Theano的高阶应用程序接口, 进行深度学习模型的设计、调试、评估、应用和可视化。

仅有B选项表示的Matplotlib不是, Matplotlib 是一个 Python 的 2D绘图库, 它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形, 不支持深度学习。

2. C

: 解析1:

本题考查python语言的用法相关问题。

在Python语言中, 是一种可变的、有序的序列结构, 其中元素可以重复。

在python中, 元组(tuple)、字符串(str)、集合(set)元素都可以重复。并不能强调是一种可变的、有序的序列结构。

而列表 (list) 是python中最基本的数据结构, 是一种有序可重复的集合, 可以随时添加和删除其中的元素。

3. B

解析1:

本题考查python相关问题。

python语义的特点:

跨平台、开源、简单易学、面向对象、可移植性、解释性、开源、高级语言、可扩展性、丰富的库、动态编程等等

综上所述B选项错误, python不是编译型语言, 而是解释型语言。

4. D

解析1:

本题考查NFA有限自动机相关问题。

针对这类问题, 可以采取找出对应反例的形式表示。

S_0 是初态, S_3 是终态, 识别出从 S_0 为初态到 S_3 为终态的路径。

可以看到无论如何到达 S_3 终态都需要经过 S_1 - S_2 , 即末尾必须存在"101"结尾的。

对于A和B选项不能包含连续字符的"0"和"1", 我们可以看到在 S_0 初态中, 有1个字符串0和1自循环, 是可以包含连续的"0"和"1"的, 所以错误。

对于C选项必须以"101"开头, 说法错误, 可以任意10的字符开头。

5. B

解析1:

考查分析语义分析阶段相关问题。

语义分析阶段主要是分析各语法结构的含义, 检查源程序是否包含静态语义错误, 并收集类型信息提供后面的代码生成阶段使用。

在确认源程序的语法和语义后, 可以对其进行翻译并给出源程序的内部表示。对于声明语句, 需要记录所遇到的符号的信息, 所以应该进行符号表的填充工作, 用来记录源程序中各个符号的必要信息, 以辅助语义的正确性检查和代码生成。

至于决策表是用于测试的, 广义表是针对数据结构的表示, 索引表是数据库中指示逻辑和物理记录对应的关系。

6. C

解析1:
 本题考查算术表达式相关问题。
 算术表达式与树的中缀表达式类似，按照左根右的顺序，其中在算术表达式中符号位表示根。
 根据该二叉树的表示，我们可以得知*为该树的总根，将左子树和右子树分隔开来。左边部分是a，右边部分是以-作为右子树的总根，左边是b/c，右边是d
 综合得出算术表达式应该为a* (b/c-d)

7. C

： 解析1:
 根据题干给出的信息，foo(5)，在该函数中，初始情况x=5，a=2。
 调用hoo(a,x)函数，此时a=2传给hoo函数中的x，并且由于是引用方式，此时x在hoo函数内的修改会影响原a的值（注意区别，hoo函数的x与foo函数的x是不用的局部变量）。同时x=5传给hoo函数中的y，并且是值调用方式，此时y在hoo函数内的修改是不会影响原x的值。
 接着分析hoo()函数，x的初始值为2，y初始值为5。执行y=y+10=15，执行x=y-x=15-2=13，此时x的值会传回原foo函数的a，即a=x=13。
 返回foo函数进行分析，输出的a和x，取值分别为修改后的13和原数值5。
 本题选择C选项。

8. AD

本题考查的是表达式和有限自动机相关内容。
 利用推导式构造语句a+b*c*d的过程如下：

整理之后的语法树，结构如下：

第一空选择A选项。
 第二空根据有限自动机判断，从初始状态开始，接收字符a可能会保持原状态不变，也可能会转入下一个状态，是不确定的，所以不确定的有限自动机。其次改自自动机识别倒数第3个字符只有a，所以无法识别bab结尾的字符串，第二空选择D选项。

9. B

解析1:
词法分析阶段处理的错误:非法字符、单词拼写错误等。
语法分析阶段处理的错误:标点符号错误、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。
静态语义分析阶段(即语义分析阶段)处理的错误:运算符与运算对象类型不合法等错误。本题选择语义错误。
目标代码生成(执行阶段)处理的错误:动态语义错误,包括陷入死循环、变量取零时做除数、引用数组元素下标越界等错误等。

10. A

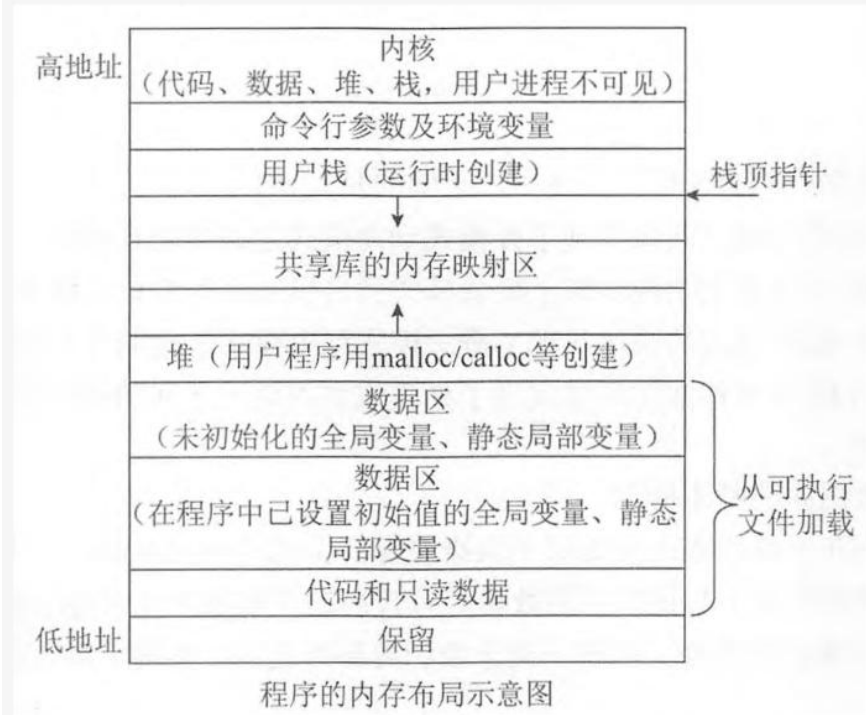
其后序遍历即为后缀表达式为: abcd/+*e-
本题选择A选项。

11. B

解析1:
本题考查程序语言基础知识。
从题中的自动机可分析出,初态q₀同时是终态,从q₀到q₀的弧(标记0)表明该自动机识别零个或多个0构成的串,路径q₀→q₁→q₀的循环表明“10”的多次重复,因此该自动机识别的字符串是“0^{*}10^{*}”的无穷多次,表示为(0|10)^{*}。

12. CD

本题考查程序语言基础知识。
程序在不同的系统中运行时,虽然对其代码和数据所占用的内存空间会有不同的布局 and 安排,但是一般都包括正文段(包含代码和只读数据)、数据区、堆和栈等。例如,在Linux系统中进程的内存布局示意图如下图所示。



栈是局部变量以及每次函数调用时所需保存的信息的存储区域,其空间的分配和释放由操作系统进行管理。每次函数调用时,其返回地址以及调用者的环境信息(例如某些寄存器)都存放在栈中。然后,在栈中为新被调用的函数的自动和临时变量分配存储空间。栈空间向低地址方向增长。
堆是一块动态存储区域,由程序员在程序中进行分配和释放,若程序语句没有释放,则程序结束时由操作系统回收。堆空间地址的增长方向是从低地址向高地址。在C程序中,通过调用标准库函数malloc/calloc/realloc等向系统动态地申请堆存储空间来存储相应规模的数据,之后用free函数释放所申请到的存储空间。

13. B

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

本题考查的是有限自动机相关知识。

A选项从s0出发, 1001到达s1, 没有到达终态s2, 不能被自动机识别。

B选项从s0出发, 1100成功到达终态s2, 可以被自动机识别。本题选择B选项。

C选项从s0出发, 1010到达s0, 没有到达终态s2, 不能被自动机识别。

A选项从s0出发, 0101到达s1, 没有到达终态s2, 不能被自动机识别。

14. A

解析1:

本题考查文法推导树相关知识。

根据本题的语法推导式, 可以发现, 这里没有终结符("("、")"、"/", 因此选项B和D错误。

在推导的过程中, 会发现""只能通过T推导, 此时必定经过了E+T或E-T, 不可能出现数字2。因此C错误。

只有A能够被推导, 推导过程如下:

(1) 通过 $E \rightarrow E-T$, 从起始符E得到E-T;

(2) 通过 $E \rightarrow E+T$, 将上面的E展开为E+T, 得到E+T-T;

(3) 通过 $E \rightarrow T \rightarrow F \rightarrow id$ →单个字母a;

(4) 通过 $T \rightarrow F \rightarrow F \rightarrow id$ →单个字母-b;

(5) 通过 $T \rightarrow F \rightarrow id$ →单个字母c。

综上, 可以从起始符E得到a+b-c, 即为A选项。

15. A

解析1:

本题考查的是函数调用过程(值调用与引用调用相关知识)。

根据题干描述的调用过程, hoo()第一个参数是传值调用, 第二个参数是引用调用, 因此, 在hoo()中对a的修改最终会影响到原foo()函数中传递的参数x, 也就是最终x打印的值。

根据hoo()函数过程, x初始传参为原args=5, 此时 $x=x-1=4$ (注意这里的x是局部变量, 只在hoo()使用), a初始传参为原x=6, 此时 $a=a*x=6*4=24$, 最终全局变量x值为24。(注意这里的原x是全局变量, 在hoo()参数中可以理解为别名为a, 现x是局部变量, 也就是之前求取的4)。

本题选择A选项。

16. C

解析1:

本题考查递归调用相关知识。

在递归调用中, 需要在前期存储某些数据, 并在后面又以存储的逆序恢复这些数据, 以提供之后使用的需求, 因此, 需要用到栈来实现递归。简单的说, 就是在前行阶段, 对于每一层递归, 函数的局部变量、参数值以及返回地址都被压入栈中。在退回阶段, 位于栈顶的局部变量、参数值和返回地址被弹出, 用于返回调用层次中执行代码的其余部分, 也就是恢复了调用的状态。本题选择C选项。

17. A

解析1:

本题考查汇编语言的执行过程。

对于编译型语言, 处理过程为: 预处理-编译-汇编-链接。

故正确答案选择A选项。

18. C

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

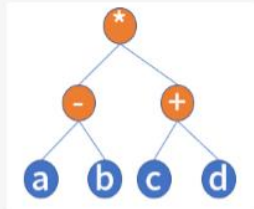
本题考查的是后缀表达式（逆波兰式）相关知识。

根据后缀表达式（逆波兰式）的定义，对图示的语法树做后序遍历即可，结果为 $abc-d^+*$ ，本题选择C选项。

19. D

解析1:

根据表达式 $(a-b) * (c+d)$ ，可以构造出语法树如下：



其后序遍历即其后缀表达式（逆波兰式）为 $ab-cd+*$ ，选择D选项。

20. AB

解析1:

A选项记号流，词法分析的输出是记号流，也就是语法分析的输入，第一空选择A选项。

B选项字符流，在Java中，根据处理的数据单位不同，分为字节流和字符流。字符流是由字符组成的，例如 File Reader、File Writer、Buffered Reader、Buffered Writer、Input Stream Reader、Output Stream Writer 等。与本题无关。

C选项源程序，词法分析的任务是把源程序的字符串转换成单词符号序列。

D选项分析树，如果没有语法错误，语法分析后就能正确的构造出其语法树。

括号不匹配是典型的语法错误，会在语法分析阶段检测出来。

21. B

22. C

解析1:

本题可以直接以实例方式排除错误选项。本题给出的NFA，能够识别字符串000，010等，以这两个字符串为例进行分析。

与之等价的DFA，也必须能够识别这样的串。A选项不能识别000，B选项不能识别010，D选项不能识别010。只有C选项能够同时识别这2个串，因此本题选择C选项。

23. A

解析1:

在词法分析阶段，其任务是从左到右逐个字符地读入源程序，对构成源程序的字符流进行扫描和分解，从而识别出一个个单词（也称单词符号或符号）。这里所谓的单词是指逻辑上紧密相连的一组字符，这些字符组合在一起才表示某一含义。词法分析过程依据的是语言的词法规则，即描述“单词”分析构成程序的字符及由字符按照构造规则构成的符号是否符合程序语言的规定”是对单词的检查。

语法分析的任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位，如“表达式”“语句”和“程序”等。语法规则就是各类语法单位的构成规则。

语义分析阶段分析各语法结构的含义，检查源程序是否包含静态语义错误，并收集类型信息供后面的代码生成阶段使用。

24. B

解析1:

符号表在编译程序工作的过程中需要不断收集、记录和使用源程序中一些语法符号的类型和特征等相关信息。这些信息一般以表格形式存储于系统中。

25. C

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

词法分析阶段依据语言的词法规则，对源程序进行逐个字符地扫描，从中识别出一个个“单词”符号，主要是针对词汇的检查。

语法分析的任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位，如“表达式”“语句”和“程序”等。语法规则就是各类语法单位的构成规则，主要是针对结构的检查。

语义分析阶段分析各语法结构的含义，检查源程序是否包含语义错误，主要针对句子含义的检查。

本题描述的是语法分析。

26. A

解析1:

程序设计语言的基本成分包括数据、运算、控制和传输等。

程序设计语言的控制成分包括顺序、选择和循环3种结构。

所以本题选择A选项。

27. D

解析1:

本题采用引用调用，会改变实参的值。对于实参a，传递给g(a)之后，在g(a)函数，表现为形参x。

根据g(x)代码：m=5*2=10，x=10-1=9，返回值x+m=19；

返回f()代码，此时a（即g(x)中的x）的值已经改变，为9；c等于g(a)的返回值，也就是19。

最终可得f()的返回值a+c=28。

28. A

解析1：本题因为是不确定的有限自动机，中间内容有多种可能，但由图可以看到，从初态0开始，首字符只能为0，到终态结束之前，尾字符也只能为0，根据选项可以排除，本题选择A选项。

29. B

解析1：词法分析：从左到右逐个扫描源程序中的字符，识别其中如关键字（或称保留字）、标识符、常数、运算符以及分隔符（标点符号和括号）等。

语法分析：根据语法规则将单词符号分解成各类语法单位，并分析源程序是否存在语法上的错误。包括：语言结构出错、if...end if不匹配，缺少分号、括号不匹配、表达式缺少操作数等。本题属于语法分析阶段的作用。

语义分析：进行类型分析和检查，主要检测源程序是否存在静态语义错误。包括：运算符和运算类型不符合，如取余时用浮点数。

30. C

：解析1：本题A选项需要将浮点数转换为整型数；B选项和D选项同类型数据，不需要转换数据类型；C选项需要将整型数转换为浮点数再计算。因此本题选择C选项。

31. C

解析1：本题由二叉树可知，表达式最后计算的为-，其次为*，最先做的为+，根据选项可知本题选C选项。

也可将A、B、C、D四个选项对应的二叉树全部画出，找出相符的选项。

32. C

解析1：翻译的任务：首先是语义分析和正确性检查，若正确，则翻译成中间代码或目标代码。其基本思想是，根据翻译的需要设置文法符号的属性，以描述语法结构的语义。例如，一个变量的属性有类型，层次，存储地址等。表达式的属性有类型，值等。属性值的计算和产生式相联系。随着语法分析的进行，执行属性值的计算，完成语义分析和翻译的任务。

33. B

解析1：一个上下文无关语法定义一个语言，其主要思想是从文法的开始符号出发，反复连续使用产生式，对非终结符进行替换和展开。

34. A

解析1：根据图中展示，其正规式应以a开头，a结尾，b可以出现0次或多次，所以是ab*a，基本上可以排除BCD三项，答案为A。

35. B

解析1：对题中ABCD4个二叉树进行后序遍历，得出结果与该后缀表达式一致的则为与其等价的语法树。答案为B。

36. C

解析1：编译语言是一种以编译器来实现的编程语言。它不像直译语言一样，由解释器将代码一句一句运行，而是以编译器，先将代码编译为机器码，再加以运行。将某一种程序设计语言写的程序翻译成等价的另一种语言的程序的程序，称之为编译程序。

37. A

解析1：LISP是一种通用高级计算机程序语言，长期以来垄断人工智能领域的应用。LISP作为因应人工智能而设计的语言，是第一个声明式系内函数式程序设计语言，有别于命令式系内过程式的C、Fortran和面向对象的Java、C#等结构化程序设计语言。

38. D

解析1：传值调用中，形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变；而引用（传址）调用中，形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值。

$x = 2 * 10 - 1 = 19;$

a是引用传递，所以 $a = a + x = 10 + 19 = 29$ ，结果为29。

39. A

解析1：B项从0到1然后走不了了，C项在3状态结束，不对；D项也只到1状态。

40. C

解析1：语法分析是编译过程的一个逻辑阶段。语法分析的任务是在词法分析的基础上将单词序列组合成各类语法短语，如“程序”，“语句”，“表达式”等等。语法分析程序判断源程序在结构上是否正确。源程序的结构由上下文无关文法描述。

41. B

：解析1：后缀式和三地址码是常用的中间代码，CD与具体的机器无关。

42. A

解析1：脚本语言中使用变量和函数来完成程序。

43. A

解析1：现行操作系统均由C/C++开发。

44. A

解析1：要得到题目中的表达式语法树后缀形式，只需要对树进行后序遍历即可，后序遍历的结果为： $x5y + a/b -$ 。

45. B

解析1：当值传递的时候，将原来的参数复制了一份，但是引用传递的时候是将变量本身传了出去，因此，a代表的其实就是x本身，f函数里面的x是另一个变量，只有a的变化才能导致main函数里面的x值的变化。

46. C

解析1：选项中，只有C选项的字符串能被DFA解析。解析路径为：ACEEBDD。

47. B

解析1：检查单个词是否正确，属于词法阶段的工作。而识别判断程序语句形式是否正确属于语法分析的工作。

48. D

解析1: 正规式(a|b)*对应的正规集为{ε, a, b, aa, ab, ..., 所有由a和b组成的字符串}, 结尾为b。

49. B

解析1: 关键字和注释不能作为标识符给对象命名。

在高级程序语言中, 程序员可以定义变量名、函数名, 也可以自定义数据类型, 比如以类似于[typedef 原数据类型 新数据类型]格式, 定义新的数据类型名。

50. D

解析1: 在本题中, 需仔细阅读代码, for语句后有“;”号, 说明该循环语句的语句体为空, 因此k在循环过程中没有进行自增操作, 此时, 整个代码会不停的进行空操作, 进入死循环, 而此时的死循环属于动态语义错误。

51. B

解析1: 上下文无关文法: 形式语言理论中一种重要的变换文法, 用来描述上下文无关语言, 在乔姆斯基分层中称为2型文法。由于程序设计语言的语法基本上都是上下文无关文法, 因此应用十分广泛。

52. A

解析1: A的方式可以保证a后面必定是b。

对于B、C选项, 当b的*取值为0时, a的后面不能保证会有b。

对于D选项, 表示的时任意a和b组成的串, 因此包括aaa, 不满足a的后面必须有b。

本题只有A选项符合题意。

53. C

解析1: 传值调用最显著的特征就是被调用的函数内部对形参的修改不影响实参的值。引用调用是将实参的地址传递给形参, 使得形参的地址就是实参的地址。

54. A

解析1:

编译程序的功能是把用高级语言书写的源程序翻译成与之等价的目標程序。编译过程划分成词法分析、语法分析、语义分析、中间代码生成、代码优化和目標代码生成6个阶段。目標程序可以独立于源程序运行。

解释程序是一种语言处理程序, 在词法、语法和语义分析方面与编译程序的工作原理基本相同, 但在运行用户程序时, 它是直接执行源程序或源程序的内部形式(中间代码)。因此, 解释程序并不产生目標程序, 这是它和编译程序的主要区别。

55. C

解析1: 脚本语言 (Script languages, scripting programming languages, scripting languages) 是为了缩短传统的编写-编译-链接-运行 (edit-compile-link-run) 过程而创建的计算机编程语言。此命名起源于一个脚本“screenplay”, 每次运行都会使对话框逐字重复。早期的脚本语言经常被称为批处理语言或工作控制语言。一个脚本通常是解释运行而非编译。

56. B

解析1: 中间代码的表达形式有语法树, 后缀式, 三地址代码。

57. D

解析1: 本题考查传址与传值的相关知识, 传值调用中, 形参取的是实参的值, 形参的改变不会导致调用点所传的实参的值发生改变; 而引用(传址)调用中, 形参取的是实参的地址, 即相当于实参存储单元的地址引用, 因此其值的改变同时就改变了实参的值。

可以使用手动执行程序的方式进行。在主函数中, 调用f(5,x)之后:

f()函数中的x=5, a=1。

x=2*x+1, 则x=11。

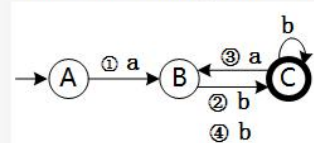
a=a+x, 则a=12。由于a是以传址的方式传入的参数, 所以主函数中的x与其值相同, 也为12。打印结果应为12。

58. B

解析1：对于该有限自动机，A为初态，C为终态，因此能识别的串一定是a开始b结束的串，可以排除C、D选项。

并且，对于该自动机能识别的串，经过初始a到达B状态后，只能识别b字符，因此A选项也错误。

本题只能选择B选项abab，识别顺序如下图所示：



59. B

解析1：归约分析是自底向上方法中的典型。先分析词，即词法分析。而分析词的组，即语法分析。

60. A

解析1：传值调用形参改变不会改变原实参的值，传址调用形参改变会改变原实参的值。

当程序执行到函数f中，调用f(x,a)时，x=5,a=16。

当程序在f(x,a)中执行完成之后，对r的值的改变，并不会影响原实参值，但对s的修改会改变调用的原实参值，在f(x,a)中执行完成之后s的值变为38，所以对应原实参a也变为38，而原实参x的值没有改变，仍然是5。最后返回值是：a-x，即38-5=33。

61. D

解析1：要证明一种说法有误只需要举一反例即可，所以做这类题时，举反例排除错误选择是一个不错的选择。

由于题目所述的NFA可以解析串“1”，所以可排除：A，B，C三个选项。

62. C

：解析1：死循环错误属于典型的语义错误，但静态的语义错误可被编译器发现，到程序真正陷入死循环说明编译器并未发现，所以属于动态语义错误。

63. A

解析1：逆波兰使用栈的基本操作流程为：从左至右将数字入栈，当遇运算符时，出栈运算符所需数据进行操作，再将操作结果入栈，依此类推。

64. CB

解析1：在编译过程中：词法分析；语法分析；语义分析；目标代码生成是必须的，而代码优化和中间代码生成是可以不需要的。

编译与解释的区别在于：

编译直接生成目标代码，在机器上执行而编译器不需要参与执行，因此程序执行速度快；

解释则生成中间代码或其等价形式，程序执行时需要解释器的参与，并且由解释器控制程序的执行，因此执行速度慢。

65. B

解析1：所谓递归下降法（recursive descent method），是指对文法的每一非终结符号，都根据相应产生式各候选式的结构，为其编写一个子程序（或函数），用来识别该非终结符号所表示的语法范畴。

66. A

解析1：本题使用代入法进行验证比较容易。

（1）代入aaa，选项B与C无法解析，故排除。

（2）代入ba，选项D无法解析，也要排除，此时可以确定正确答案为A。

67. D

解析1： 本题考查程序语言基础知识。

用某种高级语言或汇编语言编写的程序称为源程序，源程序不能直接在计算机上执行。汇编语言源程序需要用一个汇编程序将其翻译成目标程序后才能执行。高级语言源程序则需要对应的解释程序或编译程序对其进行翻译，然后在机器上运行。

解释程序也称为解释器，它或者直接解释执行源程序，或者将源程序翻译成某种中间代码后再加以执行；而编译程序（编译器）则是将源程序翻译成目标语言程序，然后在计算机上运行目标程序。这两种语言处理程序的根本区别是：在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序（或其某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序。简单来说，在解释方式下，翻译源程序时不生成独立的目标程序，而编译器则将源程序翻译成独立保存的目标程序。

68. B

解析1： 全局变量、静态局部变量、静态全局变量都存放在静态数据存储区。

69. B

解析1： 本题考查的是表达式的树形表示，我们常见的表达式形式是树的中序遍历序列。

对算术表达式 $(a+(b-c))*d$ 求值的运算处理顺序是：先进行 $b-c$ ，然后与 a 相加，最后再与 d 相乘。只有选项B所示的二叉树与其相符。

70. C

解析1： 本题考查程序语言基础知识。

选项A涉及程序语言的一般概念，程序设计语言的基本成分包括数据、运算、控制和传输等。

选项B考查高级语言和低级语言的概念。对于程序设计语言，高级语言和低级语言是指其相对于运行程序的机器的抽象程度。低级语言在形式上更接近机器指令，汇编语言就是与机器指令——对应的。高级语言对底层操作进行了抽象和封装，其一条语句对应多条机器指令，使编写程序的过程更符合人类的思维习惯，并且极大简化了人力劳动。高级语言不依赖于具体的机器硬件。

选项C考查局部变量的概念，凡是在函数内部定义的变量都是局部变量（也称作内部变量），包括在函数内部复合语句中定义的变量和函数形参表中说明的形式参数。局部变量只能在函数内部使用，其作用域是从定义位置起至函数体或复合语句体结束为止。局部变量的值通常在其生存期内是变化的。

选项D考查常量的概念，程序中常量的值在运行时是不能改变的。

71. B

解析1： 上下文无关文法：形式语言理论中一种重要的变换文法，用来描述上下文无关语言，在乔姆斯基分层中称为2型文法。由于程序设计语言的语法基本上都是上下文无关文法，因此应用十分广泛。

72. A

解析1： 1既是初态也是终态，从图中可以看出a能连续出现，保持在状态1。

73. D

解析1： 目标代码生成阶段应考虑直接影响到目标代码速度的三个问题：一是如何生成较短的目标代码；二是如何充分利用计算机中的寄存器，减少目标代码访问存储单元的次數；三是如何充分利用计算机指令系统的特点，以提高目标代码的质量。

74. D

解析1： 本题考查程序语言基础知识。

从原理上讲，对源程序进行语义分析之后就可以直接生成目标代码，但由于源程序与目标代码的逻辑结构往往差别很大，特别是考虑到具体机器指令系统的特点，要使翻译一次到位很困难，而且用语法制导方式机械生成的目标代码往往是繁琐和低效的，因此有必要设计一种中间代码，将源程序首先翻译成中间代码表示形式，以利于进行与机器无关的优化处理。由于中间代码实际上也起着编译器前端和后端分水岭的作用，所以使用中间代码也有助于提高编译程序的可移植性。常用的中间代码有后缀式、三元式、四元式和树（图）等形式。

75. A

解析1： 后缀表达式是从左到右算。把表达式加上括号， $((a-b)*(c+d))$ ，再把运算符加到括号外面 $ab-cd+*$ 。答案为A。

76. B

解析1:

C++: 是在C语言的基础上发展起来的, 主要增加了类的功能, 使其成为面向对象的程序设计语言。

Prolog: 是以特殊的逻辑推理形式回答用户的查询, 它建立在关系理论和一阶谓词理论基础上。经常用于数据库和专家系统。

Python: 是一种面向对象、解释型计算机程序设计语言, 由Guido van Rossum于1989年底发明, 第一个公开发行版发行于1991年。Python语法简洁而清晰, 具有丰富和强大的类库。它常被昵称为胶水语言, 它能够把用其他语言制作的各种模块 (尤其是C/C++) 很轻松地联结在一起。

C语言: 编译型语言。

77. D

解析1: 检查语法错误是在编译时, 表达式“c=b/a”符合语法逻辑, 编译时不会报语法错误; 由于编译时a的值无法确定, 需要到运行时, 实际传入值时才能确定, 因此在运行时, 若分母为0, 将产生异常。

78. B

解析1:

上下文无关文法重要的原因在于它们拥有足够强的表达力来表示大多数程序设计语言的语法; 实际上, 几乎所有程序设计语言都是通过上下文无关文法来定义的。另一方面, 上下文无关文法又足够简单, 使得我们可以构造有效的分析算法来检验一个给定字符串是否是由某个上下文无关文法产生的。

79. A

解析1:

编译程序的功能就是把高级语言书写的源程序翻译成与之等价的目标程序 (汇编语言或机器语言)。

解释程序是另一种语言处理程序, 在词法、语法和语义分析方面与编译程序的工作原理基本相同, 但在运行时直接执行源程序或源程序的内部形式, 即解释程序不产生源程序的目标程序, 这点是它与编译程序的主要区别。

80. A

解析1:

符号表: 符号表是一种用于语言翻译器 (例如编译器和解释器) 中的数据结构。在符号表中, 程序源代码中的每个标识符都和它的声明或使用信息绑定在一起, 比如其数据类型、作用域以及内存地址。

哈希表: 也叫散列表, 是根据关键码值 (Key value) 而直接进行访问的数据结构。也就是说, 它通过把关键码值映射到表中一个位置来访问记录, 以加快查找的速度。

动态查找表: 动态查找表的表结构本身是在查找过程中动态生成的, 即对于给定值key, 若表中存在其关键字等于key的记录, 则查找成功返回, 否则插入关键字等于key的记录。

栈和队列: 基本的数据结构。栈的基本特点是“后进先出”, 而队列的基本特点是“先进先出”。

81. B

解析1:

在函数调用时, 系统为形参准备空间, 并把实参的值赋值到形参空间中, 在调用结束后, 形参空间将被释放, 而实参的值保持不变, 这就是传值传递方式。传值传递方式中实参与形参之间的数据传递是单向的, 只能由实参传递给形参, 因而即使形参的值在函数执行过程中发生了变化, 也不会影响到实参值。在C语言中, 当参数类型是非指针类型和非数组类型时, 均采用传值方式。

传地址方式把实参的地址赋值给形参, 这样形参就可以根据地址值访问和更改实参的内容, 从而实现双向传递。当参数类型是指针类型或数组类型时, 均采用传地址方式。

82. C

解析1:

HTML: 静态网页

LISP: 一种基于λ演算的函数式编程语言

PHP: 混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。

83. C

解析1:

强、弱类型语言有两种划分方法。

一种是从内存上来说:允许将一块内存看做多种类型。比如直接将整型变量与字符变量相加。

另一种是从语法上来说:在变量或者对象使用前,需要声明类型的语言就是强类型语言。

根据题意对弱语言的定义:不需要进行变量/对象类型声明的语言,可知本题是从语法的角度来区分强弱类型语言。java、c/c++、c#中变量在使用前都需要声明其类型,而python则不需要。因此从语法上来说,python是弱类型语言。

84. A

解析1:

编译程序的功能是从源代码(通常为高级语言)到能直接被计算机或虚拟机执行的目标代码(汇编语言或机器语言)的翻译过程。工作过程分为6个阶段:词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成。

各个阶段逻辑上可以分为前端和后端。前端主要负责解析输入的源代码,由语法分析器和语义分析器协同工作。语法分析器负责把源代码中的“单词”找出来,语义分析器把这些分散的单词按预先定义好的语法组装成有意义的表达式、语句、函数等等。前端还负责语义的检查,例如检测参与运算的变量是否是同一类型的,简单的错误处理。最终的结果常常是一个抽象的语法树,这样后端可以在此基础上进一步优化处理。

后端编译器后端主要负责分析,优化中间代码以及生成机器代码。

有限自动机是进行词法分析的工具。

85. B

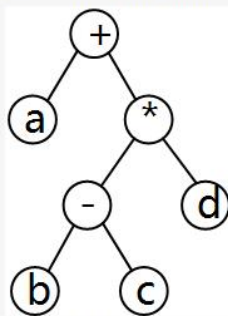
解析1:

中间代码是源程序的一种内部表示,或称中间语言。中间代码的作用是可使编译程序的结构在逻辑上更为简单明确,特别是可使目标代码的优化比较容易实现中间代码,即为中间语言程序,中间语言的复杂性介于源程序语言和机器语言之间。中间语言有多种形式,常见的有逆波兰记号、四元式、三元式和树。

86. B

解析1:

本题要求算术表达式的后缀式,解决该类问题的方法是将算术表达式构造成一棵二叉树,然后对二叉树进行后序遍历,得到后缀式。题目中算术表达式可以构造为以下二叉树:



对该二叉树进行后序遍历结果为: a b c - d * +。

87. C

解析1:

对于文法可推导出的字符串分析,考试一般可对文法举例,然后总结规律。

以本文文法为例,可以产生的字符串包括:

(1) 10

推导过程: $S \rightarrow A0$; $A \rightarrow 1$ 。

(2) 01

推导过程: $S \rightarrow B1$; $B \rightarrow 0$ 。

(3) 1010

推导过程: $S \rightarrow A0$; $A \rightarrow S1$: $S \rightarrow A0, A \rightarrow 1$ 。

至此,可以了解到,选项A、B、D的描述都是不正确的。

88. A

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

程序运行时，对函数的调用一般有两种形式：传值调用和引用调用。

传值调用：形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变。

引用（传址）调用：形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值。

89. AC

解析1:

编译程序的功能是从源代码（通常为高级语言）到能直接被计算机或虚拟机执行的目标代码（汇编语言或机器语言）的翻译过程。工作过程分为6个阶段：词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成。

各个阶段逻辑上可以分为前端和后端。前端主要负责解析输入的源代码，由语法分析器和语义分析器协同工作。语法分析器负责把源代码中的单词找出来，语义分析器把这些分散的单词按预先定义好的语法组装成有意义的表达式、语句、函数等等。前端还负责语义的检查，例如检测参与运算的变量是否是同一类型的，简单的错误处理。最终的结果常常是一个抽象的语法树，这样后端可以在此基础上进一步优化处理。

后端编译器后端主要负责分析，优化中间代码以及生成机器代码。

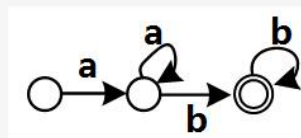
对于声明语句，需要记录符号信息，因此应该填入合理组织的符号表，而对于可执行语句，需要翻译成中间代码或目标代码。

90. D

解析1:

与L语言最接近的正规式为： aa^*bb^*

与L语言最接近的有限自动机为：



但无论如何无法构造出完全等价的正规式，因为在L语言中， n 的取值无论为多少， a 与 b 的数量都相等，而 aa^*bb^* 无法保障这一点。

91. A

解析1:

在编译时产生的目标文件都是从地址0开始的，连接时将各个目标文件进行符号替换，这时会修改相应的地址，最后产生一个从地址0开始的可执行文件。在该可执行文件中的地址称为逻辑地址。

92. C

解析1:

传值调用最显著的特征就是被调用的函数内部对形参的修改不影响实参的值。引用调用是将实参的地址传递给形参，使得形参的地址就是实参的地址，函数中对形参的修改会导致原实参值的改变，因此可以实现实参与形参之间数据的双向传递。

在传值调用中，实参可以是变量，也可以是常量和表达式，而在传址调用中，实参不能是常量。

93. C

解析1:

编译程序：将源程序编译成可以直接运行的目标语言程序。解释程序：直接解释执行的源程序或者将源程序翻译成某种中间代码后再加以执行，解释程序需要参与运行。

94. A

95. D

解析1:

高级程序语言编写的程序要想在计算机上执行，就需要翻译，常见的翻译方式有编译和解释两种。在编译的方式下，经常用一些与之等价的中间代码表示形式，而常用的中间代码有树、后缀式、四元式。

96. B

2022 知识点练习->上午->10.程序设计语言答案解析

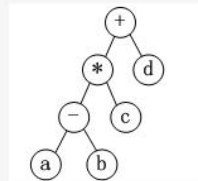
解析1:

用高级语言编写的程序不能被机器执行，而是经过相关的处理后，将其变成目标程序后才能被机器执行，而这一过程就是一个编译的过程，而实现这个功能的程序就称为编译程序。

97. C

解析1:

本题要求通过中缀表达式，求后缀式（也称为逆波兰）。解答这类问题，可以借助于二叉树。因为中缀表达式对应于一颗二叉树的中序遍历，前缀表达式对应于二叉树的前序遍历，后缀表达式对应于二叉树的后序遍历。所以在本题中，需要先把二叉树构造处理。将表达式 $(a-b)*c+d$ 构造二叉树，如图所示。



将此树进行后序遍历，得到： $ab-c*d+$ 。

98. B

解析1:

由于m的值大于等于0，因此a的个数可能没有，也可能是多个，与之对应的是 a^* ，而n的值是大于等于1的，因此至少有一个b，那么与之对应的就是 bb^* ，所以本题答案选B。

99. A

解析1:

Python是一种面向对象、直译式计算机程序设计语言，它不仅可以编写独立的程序，还常被分类为“脚本语言”。

100. C

解析1:

编译和解释是语言处理的两种基本方式。编译过程包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等阶段，以及符号表管理与出错处理模块。

解释过程在词法、语法和语义分析方面与编译程序的工作原理基本相同，但是在运行用户程序时，它直接执行源程序或源程序的内部形式。

这两种语言处理程序的根本区别是：在编译方式下，机器上运行的是与源码程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序（或其某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序。

在编译方式下，词法、语法和语义分析是必须要进行的工作，而生产中间代码和优化则是可以进行也可以不进行。

101. D

解析1:

形式参数就是过程定义中函数名后括号中所带的参数；实际参数是在调用点表示向被调用过程传递的数据。

在函数调用时，数据传递的方向是从实参到形参。只是采用传值传递方式时，传递的是数值，这个数值只要是确定的即可，可以是常量、变量或表达式等。而采用传址传递方式时，传递的是地址，因此实参必须有地址。

102. C

解析1:

被有限自动机所识别是指从初态开始到终态结束，所输入的字符串能够按顺序地执行下去，若到某个状态不能往下走得到下一个字符，则认为不能识别。

在本题中，从初态A出发，不管经过多少个1和0之后，只能是处在A、B、C三种状态中的一种，所以在 $(0|1)^*$ 后，只能是处在A、B、C三种状态中的一种，不管是在那个状态，输入0后，都会处在状态B，然后输入1，都会转换到状态C，因此与本题有限自动机等价的正规式是 $(0|1)^*01$ ，即该自动机所识别的字符串的特点是必须以01结尾的0、1串。

103. D

解析1:

后缀表示也称为表达式的逆波兰表示。在这种表示方法中，将运算符号写在运算对象的后面，表达式中的运算符按照计算次序书写。

对于表达式 $x-(y+c)*8$ ，先计算 y 与 c 的和，再乘以8，最后用 x 减去这个这个计算，因此其后缀式为 $xy c + 8 * -$ 。

104. BD

解析1:

传值调用中，形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变；而引用（传址）调用中，形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值。本题中 a 是全局变量，因此会随着操作发生值的改变。

在本题中，首先是采用传值调用，这个时候将变量 a 的值5传递给形参 r ，即 r 的值为5，那么 a 的值经过 $a=r+1$ 后变成了6（ r 的值不变），而 r 的值经过 $r=r*2$ 后变成了10，并返回，即在函数 t 中，变量 x 的值被赋值为10，那么在函数 t 中最后输出的时 $10+6=16$ 。

采用引用调用时，由于形参 r 指向的是实参 a 的存储空间，即 r 与 a 指向的是同一块存储单元，首先 a 的值为5，经过 $a=r+1$ 后变成了6（ r 的值变为6），再经过 $r=r*2$ 后变成了12，并返回，即在函数 t 中，变量 x 的值被赋值为12，那么在函数 t 中最后输出的时 $12+12=24$ 。

105. A

解析1:

被有限自动机所识别是指从初态开始到终态结束，所输入的字符串能够按顺序地执行下去，若到某个状态不能往下走得到下一个字符，则认为不能识别。

在本题中，选项A能被识别。从初态A出发，不管经过多少个1和0之后，只能是处在A、B、C三种状态中的一种，所以在 $(0|1)^*$ 后，只能是处在A、B、C三种状态中的一种，不管是在那个状态，输入0后，都会处在状态B，然后输入1，都会转换到状态C，因此选项A能被该有限自动机所识别。

同样的道理，我们可以知道其它选项的正规式不能被识别。

106. B

解析1:

要求源程序中的数据必须具有类型的目的主要有以下几个方面:

第一是方便为数据合理分配存储单元;第二是规定了数据类型,就知道了其占用的字节数,从而也就规定了数据对象的取值范围及能够进行的运算;第三是对参与表达式求值的数据对象可以进行合法性检查,比如浮点数就不能进行自加操作。

107. D

解析1:

在C程序中，若在某个表达式中引用了未赋初值的变量，那么程序是可以编译并运行的，因为程序中并没有语法方面的错误，只是运行的结果可能与期望的结果不一致。

108. C

解析1:

在函数调用时，系统为形参准备空间，并把实参的值赋值到形参空间中，在调用结束后，形参空间将被释放，而实参的值保持不变，这就是传值传递方式。传值传递方式中实参与形参之间的数据传递是单向的，只能由实参传递给形参，因而即使形参的值在函数执行过程中发生了变化，也不会影响到实参值。在C语言中，当参数类型是非指针类型和非数组类型时，均采用传值方式。

传地址方式把实参的地址赋值给形参，这样形参就可以根据地址值访问和更改实参的内容，从而实现双向传递。当参数类型是指针类型或数组类型时，均采用传地址方式。

区别于参数传值方式和返回值传递方式，传地址方式具有以下明显的优势。

(1) 参数传值方式是主调函数与被调函数之间的单向数据传递方式，而参数的传地址方式则实现了二者之间的双向数据传递。

(2) 函数的返回值每次只能把一个数据项从被调函数传递到主调函数，而参数的传地址方式却可一次性地传递多个数据项到主调函数。

109. C

解析1:

本题主要考查有限自动机。

在本题中，A是初始状态，C是终止状态，通过选项中的字符串可以从初始状态到达终止状态，则说明该字符串能被题目中的自动机识别。也可以理解为依次输入选项中的字符串，可以在该自动机中找到相应的路径。

对于选项A的字符串0000，在输入0后，从初始状态A转移到状态B，然后接着输入3个0，状态然后停留在B，而无法到达终态C，因此选项A不能被该自动机识别。

同样的道理，我们可以找到字符串0101能被该自动机识别，在输入0后，状态跳转到B，输入1则由B转至C，再输入0，又由C转至B，最后输入1，由B转至终态C。

110. D

解析1:

在对用高级程序设计语言编写的程序进行执行时，首先是将源代码翻译成目标代码，然后在连接成可执行的二进制代码。因此在翻译阶段，目标代码生成阶段的工作与目标机器的体系结构密切相关。

111. B

解析1:

一种程序设计语言规定其程序中的数据必须具有类型，好处如下：

(1) 有利于在翻译程序的过程中为数据合理分配存储单元，因为程序设计语言为不同的数据类型规定了其所占的存储空间，如果数据类型确定，其所占的存储空间也是确定的。

(2) 有利于对参与表达式计算的数据对象进行检查，因为知道数据的数据类型，我们就可以根据类型来判断该数据是否可以参与某表达式计算，如自加、自减的操作数不允许是浮点数，这只要根据数据的类型就能判断某操作数，是否能进行自加、自减运算。

(3) 有利于规定数据对象的取值范围及能够进行的运算，根据数据类型，我们可以知道数据的存储空间，也同时能知道数据的表示范围，如C语言中的整型数据，它占两个字节（16位），能表示的数据范围就是 -2^{16} 至 $2^{16}-1$ 。

综上所述，可知本题的正确答案选B。

112. BC

: 解析1:

逆波兰式也叫后缀表达式，即将运算符写在操作数之后的表达式，它不需使用括号，在将算术表达式转换为逆波兰式表示时，需要分配2个栈，一个作为临时存储运算符的栈S1（含一个结束符号），一个作为输入逆波兰式的栈S2（空栈）。

而逆波兰式 $ab-cd+*$ 转换为中缀表达式的过程为： $ab-cd+* = (ab-)*(cd+) = (a-b)*(cd+) = (a-b)*(c+d)$ 。因此本题答案选C。

113. A

解析1:

本题中给出的是一个双重循环结构，循环体就是 $count++$ 。第一层循环的循环次数为4次，分别为 $i=1, 2, 4, 8$ 的情况。而当 $i=1$ 时，第二层循环循环1次；当 $i=2$ 时，第二层循环2次；当 $i=4$ 时，第二层循环4次；当 $i=8$ 时，第二层循环8次。那么可知循环体一共执行了 $1+2+4+8=15$ 次。

114. B

解析1:

面向机器的程序设计语言，使用汇编语言编写的程序，机器不能直接识别，要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序。汇编程序输入的是用汇编语言书写的源程序，输出的是用机器语言表示的目标程序。

115. A

解析1:

可视化程序设计主要是让程序设计人员利用软件本身所提供的各种控件，像搭积木式地构造应用程序的各种界面。可视化程序设计最大的优点是设计人员可以不用编写或只需编写很少的程序代码，就能完成应用程序的设计，这样就能极大地提高设计人员的工作效率。在可视化程序设计中，可随时查看程序的运行效果。

116. D

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

本题主要考查确定有限自动机与非确定有限自动机的判断。

非确定有限状态自动机与确定有限状态自动机的最大区别是它们的转移函数不同。确定有限状态自动机对每一个可能的输入只有一个状态的转移。非确定有限状态自动机对每一个可能的输入可以有多个状态转移，接受到输入时从这多个状态转移中非确定地选择一个。

在本题中给出的图M1中，我们可以看到当在状态A输入0时，它可以转移到它自己，也可以转移到状态B，所以M1是非确定的。而M2中不存在这样的情况，因此是确定的有限自动机。

117. B

解析1:

本题主要考查有限自动机。

在题目中，0是初始状态，3是终止状态，通过选项中的字符串可以从初始状态到达终止状态，则说明该字符串能被题目中的自动机识别。也可以理解为依次输入选项中的字符串，可以在该自动机中找到相应的路径。

对于选项A的字符串abab，通过ab可以达到终止状态，然后输入a任然可以有路径，但再输入b时，没有路径与其对应。因此A不可被该自动机识别。同样的道理，我们可以找到字符串aaaa能被该自动机识别。

118. A

解析1:

编译程序分析源程序的阶段依次词法分析、语法分析、语义分析。

119. B

解析1:

本题主要考查我们对常量与变量的理解。顾名思义，常量是指值一旦确定后就不能再变的量，而变量则是一个在程序执行过程中，可以根据需要修改的量，是一个可改变的量。当然不管是常量还是变量，它们都有其类型属性。

120. B

解析1:

本题考查程序语言基础知识。

闭包运算符"*"将其运算对象进行若干次连接，因此0*表示若干个0构成的串，而(10*1)*则表示偶数个1构成的串。

121. D

: 解析1:

本题考查程序语言基础知识。

HTML (Hypertext Marked Language超文本标记语言)，用于互联网的信息表示。用HTML编写的超文本文档称为HTML文档，它能独立于各种操作系统平台（如UNIX Windows等）。HTML文档是纯文本文档，可以使用记事本、写字板等编辑工具来编写HTML文件，其文件（文档）的扩展名是.html或.htm，它们需要通过WWW浏览器进行解释并显示出效果。

XML (Extensible Markup Language可扩展的标记语言) 1.0标准于1998年2月10日发布，被认为是继HTML和Java编程语言之后的又一个里程碑式的Internet技术。XML丰富了HTML的描述功能，可以描述非常复杂的Web页面，如复杂的数学表达式、化学方程式等。XML的特点是结构化、自描述、可扩展和浏览器自适应等。

用于WAP的标记语言就是WML (Wireless Markup Language)，其语法跟XML一样，是XML的子集。

PHP (Hypertext Preprocessor) 是一种在服务器端执行的、嵌入HTML文档的脚本语言，其语言风格类似于C语言，被网站编程人员广泛运用。

122. C

解析1:

本题考查语言处理基础知识。

编译和解释是语言处理的两种基本方式。编译过程包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等阶段,以及符号表管理与出错处理模块。

解释过程在词法、语法和语义分析方面与编译程序的工作原理基本相同,但是在运行用户程序时,它直接执行源程序或源程序的内部形式。

这两种语言处理程序的根本区别是:在编译方式下,机器上运行的是与源程序等价的目标程序,源程序和编译程序都不再参与目标程序的执行过程;而在解释方式下,解释程序和源程序(或其某种等价表示)要参与到程序的运行过程中,运行程序的控制权在解释程序。解释器翻译源程序时不产生独立的目标程序,而编译器则需将源程序翻译成独立的目标程序。

123. B

解析1:

本题考查软件程序设计的基础知识。

程序的三种基本控制结构是顺序结构、选择结构和重复结构。

124. A

解析1:

本题考查程序语言基础知识。

对C源程序进行编译时,需建立符号表,其作用是记录源程序中各个符号(变量等)的必要信息,以辅助语义的正确性检查和代码生成,在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的建立可以始于词法分析阶段,也可以放到语法分析和语义分析阶段,但符号表的使用有时会延续到目标代码的运行阶段。

125. D

解析1:

本题考查逻辑表达式的计算及程序语言处理基础知识。

“逻辑与运算”的优先级高于“逻辑或运算”。

“逻辑与运算”表达式“ $x \wedge y$ ”的短路求值逻辑是:若 x 为假,则可知“ $x \wedge y$ ”的值为假,无需再对 y 求值,因此只有在 x 为真时继续对 y 求值。

“逻辑或运算”表达式“ $x \vee y$ ”的短路求值逻辑是:若 x 为真,则可知“ $x \vee y$ ”的值为真,无需再对 y 求值,因此只有在 x 为假时继续对 y 求值。

对于逻辑表达式“ $a \wedge b \vee c \wedge (b \vee x > 0)$ ”,从运算符的优先级方面考虑需先对“ $a \wedge b$ ”求值,然后对“ $c \wedge (b \vee x > 0)$ ”求值,最后进行“ \vee ”运算,因此后缀式为“ $ab \wedge cbx0 > \vee \wedge \vee$ ”。

126. D

解析1:

本题考查程序错误的基本知识。

一般程序中的错误可分为语法错误和语义错误(或逻辑错误),语法错误是语句形式上的错误,语义错误是含义上的错误(逻辑错误)。

在C程序代码中,

```
int i=0;
while (i<10) ;
{ i=i+1; }
```

由于在while的循环条件表达式后加了分号“:”,使循环体为空语句,因此造成死循环,这是一种运行中出现的逻辑错误,属于动态语义错误。

127. B

解析1:

本题考查表达式的表示方式。

表达式的逆波兰表示也就是后缀表示,在表达式的这种表示方法中,将运算符号写在运算对象的后面,并指明其前面的操作数或中间结果所要执行的运算。对后缀表达式从左到右求值,则每当扫描到一个运算符号时,其操作数是最近刚得到的。因此“ $ab+c*d$ ”表示:先将 a 与 b 相加,然后作一元“-”运算,结果再与 c 相乘,乘运算的结果再与 d 相减,因此中缀表达式的形式为“ $-(a+b)*c-d$ ”。

128. B

解析1:

本题考查语言结构的文法表示。

推导就是用产生式的右部替换产生式左部符号。从文法的开始符号出发,不能推导出“(a, f)”、“(abc)”、“(c, (da))”,产生字符串“(fac, bb), g)”的最左推导过程如下:

$$\begin{aligned} S &\Rightarrow (S, M) \Rightarrow ((S, M), M) \Rightarrow ((M, M), M) \Rightarrow ((MP, M), M) \Rightarrow ((MPP, M), M) \\ &\Rightarrow ((PPP, M), M) \Rightarrow ((fPP, M), M) \Rightarrow ((faP, M), M) \Rightarrow ((fac, M, M) \Rightarrow ((fac, MP), M) \\ &\Rightarrow ((faa, PP), M) \Rightarrow ((fac, bP), M) \Rightarrow ((fac, bb), M) \Rightarrow ((fac, bb), g) \end{aligned}$$

129. A

解析1:

本题考查程序语言的基本成分。

程序设计语言的语法是语言的外观。给出语言的语法意味着给出语句、声明和其他语言结构的书写规则。语义则表示不同的语法结构的含义。在程序语言的手册中,语言的描述都是围绕着语法结构展开的。通常,先给出各种语句结构的语法,然后给出对应该结构的语义以描述内在含义。语用是关于程序与使用者之间的关系。

在高级程序设计语言中,语句用于描述程序中的运算步骤、控制结构及数据传输。

130. D

解析1:

本题考查程序语言基本知识。

函数式程序设计的数据结构本质上是表,而函数又可以作为值出现在表中,因此函数式程序的控制结构取决于函数,以及函数的定义和调用。函数式语言主要用于符号数据处理,如微分和积分演算、数理逻辑、游戏推演以及人工智能等其他领域。

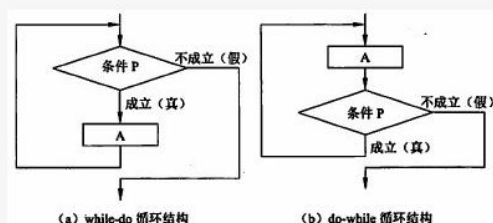
用逻辑式程序设计语言编写程序不需要描述具体的解题过程,只需要给出一些必要的事实和规则。这些规则是解决问题的方法的规范说明,根据这些事实和规则,计算机利用谓词逻辑,通过演绎推理得到求解问题的执行序列。这种语言主要用在人工智能领域,也应用在自然语言处理、数据库查询、算法描述等方面,尤其适合于作为专家系统的开发工具。

131. D

解析1:

本题考查程序语言语句的语义。

While循环也称为当型循环,其逻辑如下所示:



while语句的循环体执行次数比循环条件的判断次数少1,而do-while语句的循环体执行次数等于循环条件的判断次数。

132. B

解析1:

本题考查函数调用时的参数传递问题。

采用传值方式时,是将实际参数的值传递给形式参数,对形式参数值的修改不会影响实际参数。

采用引用方式时,是将实际参数的地址传递给形式参数,对形式参数进行修改,等同于是对实际参数进行修改。

133. B

解析1:

本题考查程序语言特点。

函数是一种对应规则(映射),它使定义域中每个元素和值域中唯一的元素相对应。函数式语言是一类以λ演算为基础的语言,其代表为LISP,主要用于人工智能领域。

逻辑型语言是一类以形式逻辑为基础的语言,其代表是建立在关系理论和一阶谓词理论基础上的PROLOG。PROLOG有很强的推理功能,适用于书写自动定理证明、专家系统和自然语言理解等问题的程序。

134. A

解析1:

本题考查程序语言基础知识。

推导就是从文法的开始符号S出发,反复使用产生式,将产生式左部的非终结符替换为右部的文法符号序列〔展开产生式用功表示),直到产生一个终结符的序列时为止。从文法G的开始符号出发,能推导出的终结符号序列(句子)的全体称为文法G产生的语言。

对于上下文无关文法 $S \rightarrow 11 \mid 1001 \mid S0 \mid SS$,从S出发可推导出11, 1001, 110, 1111, 11110和10010等,将这些二进制序列转换成对应的十进制数可知,它们都能被3整除。

135. D

解析1:

本题考查程序语言翻译基础知识。

编译程序的功能是把某高级语言书写的源程序翻译成与之等价的目标程序(汇编语言或机器语言)。编译程序的工作过程可以分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个阶段,以及出错处理和符号表管理。其中,中间代码生成和代码优化阶段不是必需的。目标代码生成时才考虑与具体机器相关的处理,寄存器分配处于代码生成阶段的工作。

136. D

解析1:

本题考查程序语言基础知识。

程序语言的语法可由上下文无关文法表示,合法的程序可看作是由该文法推导得到。

对于文法G[S],从S出发推导出[a, a]和a的过程可表示为:

$$S \Rightarrow [L] \Rightarrow [L, S] \Rightarrow [S, S] \Rightarrow [a, S] \Rightarrow [a, a]$$

$$S \Rightarrow a$$

从S出发可推导出以a或【开始的字符串,而FIRSTS(S)的成员包括a、[、]和,。

137. D

解析1:

本题考查程序语言处理基础知识。

表达式的后缀表示是指将运算符写在运算对象的后面,表达式中的运算符按照计算次序书写。

对于表达式 $(a-b) * (c+5)$,先计算a与b的差和,再计算c与5之和,最后进行相乘,因此其后缀式为 $ab-c5+*$ 。

138. B

解析1:

本题考查程序的控制结构。

do-while语句的形式为:

do

循环体语句;

while循环条件;

首先执行一次循环体语句(无条件地),若循环条件不成立,则结束do-while语句的执行;否则再一次执行循环体语句,即此后循环条件每成立一次,循环体语句就执行1次。显然,do-while中的循环体语句至少执行1次。

while语句的形式为:

while循环条件

循环体语句;

其执行过程为:先判断循环条件,若不成立,则结束While语句的执行;否则执行一次循环体语句。此后,循环条件每成立一次,循环体语句就执行1次。显然,while中的循环体语句可能一次也不执行。

因此,在不改变循环体的情况下,可用while语句来代替do-while语句,反之则不行。

139. A

2022 知识点练习->上午->10.程序设计语言答案解析

解析1:

本题考查程序翻译的基础知识。

编译程序对高级语言源程序进行翻译时，在逻辑地址空间中为变量分配存储单元，当程序开始运行时，再转换为实际的内存地址（既物理地址）。

140. C

解析1:

本题程序语言翻译基础知识。

翻译高级语言源程序的第一步工作是进行词法分析，即将源程序中的单词（记号）识别出来，该过程可用有限自动机描述。

自动机M识别一个字符串的过程是从开始状态出发，根据字符串中的字符依次进行状态转移，若能到达终态且字符串结束，则该字符串可被自动机M识别。考查题目中的选项，3857的识别过程是状态0→状态1→状态1→状态1，状态1不是终态；字符串1.2E+5中的“+”不能识别。字符串0.576E10的识别过程是状态0→状态1→状态5→状态6→状态6，在状态6下不能识别E。字符串_123.67的识别过程是状态0→状态4→状态1→状态1→状态1→状态5→状态6→状态6，因此该字符串可被题中的自动机识别。

141. B

解析1:

本题考查程序语言翻译基础知识。

非确定有限自动机NFA是一个五元组（5-tuple）； $M = (S, \Sigma, move, s_0, F)$

其中，①S是有限个状态（state）的集合；② Σ 是有限个输入字符（包括 ϵ ）的集合；③move是一个状态转移函数， $move(s_i, ch) = s_j$ 表示，当前状态 s_i ；下若遇到输入字符ch，则转移到状态 s_j ；④ s_0 是唯一的初态（也称开始状态）；⑤F是终态集（也称接受状态集），它是S的子集，包含了所有的终态。

确定的有限自动机DFA是NFA的特例：①DFA没有状态具有。状态转移（ ϵ -transition），即状态转换图中没有标记 ϵ 的边；②对每一个状态s和每一个字符a，最多有一个下一状态。

若两个FA识别同一个正规集，则这两个FA等价。对于每个NFA，都存在与之等价的DFA。

142. C

解析1:

本题考查程序设计基础知识。关于脚本语言的一些基本知识如下：

①脚本语言（JavaScript，VBScript等）是介于HTML和C、C++、Java、C#等编程语言之间的程序设计语言。HTML通常用于格式化和链接文本，而编程语言通常用于向机器发出一系列复杂的指令。

②脚本语言中也使用变量和函数，这一点与编程语言相似。与编程语言之间最大的区别是编程语言的语法规则更为严格和复杂。

③脚本语言一般都有相应的脚本引擎来解释执行，是一种解释性语言，一般需要解释器才能运行。

④脚本语言一般以文本形式存在，类似于一种命令。

下面举例说明脚本语言。设有一个可执行程序open_aa.exe，用于打开扩展名为“.aa”的文件。编写“.aa”文件需要指定一套规则（语法），open_aa.exe就用这种规则来理解文件编写人的意图并做出回应。因此，这一套规则就是脚本语言。

汇编语言是符号化的机器语言，一般情况下，用汇编语言编写的程序比高级语言效率更高。根据脚本语言的以上特点，“采用脚本语言编程可获得更高的运行效率”是错误的。

143. B

解析1:

本题考查程序语言翻译基础知识。

语言语法的一种表示法称为文法，常用的文法是上下文无关文法。

一个上下文无关文法包含以下4个部分:

①一个记号集合，称为终结符集。

②一个非终结符号集合。

③一个产生式集合。每个产生式具有一个左部和右部，左部和右部由肩头连接，左部是一个非终结符，右部是记号和（或）非终结符序列。

④一个开始符号。开始符号是一个指定的非终结符。

利用产生式产生句子的过程，是将产生式 $A \rightarrow \gamma$ 的右部代替文法符号序列 $\alpha A \beta$ 中的A得到 $\alpha \gamma \beta$ 的过程，称为 $\alpha A \beta$ 直接推导出 $\alpha \gamma \beta$ ，记作: $\alpha A \beta \Rightarrow \alpha \gamma \beta$

从S出发进行推导的过程可表示如下:

$S \Rightarrow 0S0 \Rightarrow 00S00 \Rightarrow 000S000 \Rightarrow \dots \Rightarrow 0^n 10^n$

144. C

解析1:

本题考查程序语言翻译基础知识。

编译器对高级语言源程序的处理过程可以分为词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成等阶段，以及符号表管理模块和出错处理模块。在编译器的分析综合模式中，前端将源程序翻译成一种中间表示，后端根据这个中间表示生成目标代码。目标语言的细节尽可能限制在后端。尽管可以将源程序直接翻译成目标语言代码，但使用与机器无关的中间表示形式具有以下优点:

①重置目标比较容易。不同机器上的编译器可以在已有前端的基础上附加一个适合这个机器的后端来生成。

②可以在中间表示上应用与机器无关的代码优化器。

145. A

解析1:

本题考查程序语言方面的基础知识。

一个文法的语言是该文法能产生的句子的集合。一个文法产生的句子是从文法开始符号出发推导出的所有终结符号串。

146. B

解析1:

本题考查程序语言方面的基础知识。

在正规式中，符号 $*$ 表示重复若干次（包括0次），因此正规式 $(a^*a)^*b^*$ 中的表达式 $(a^*a)^*$ 不能保证有偶数个a。同理， $(a^*(ba^*)^*b)^*$ 和 $(a|b)^*(aa)^*$ 中对a的个数也没有限制，而在 $(ab^*a)^*$ 中可以确保a的出现为偶数个。

147. D

解析1:

本题考查程序语言方面的基础知识。

对于题中自动机的状态图，先忽略状态q0的自环（识别若干个0），从初态q0到终态q1，该自动机可识别的字符串为1、101、10101、...，显然，该自动机识别的0、1串中1不能连续出现。

148. C

解析1:

本题考查程序语言基础知识。

动态语言是指程序在运行时可以改变其结构，例如新的函数可以被引进、已有的函数可以被删除等在结构上的变化等。动态语言的类型检查是在运行时进行的，其优点是方便阅读，不需要写非常多的与类型相关的代码；缺点是不方便调试，命名不规范时会读不懂、不利于理解等。

脚本语言代表一套与系统程序设计语言不同的协定。它们牺牲执行速度和与系统程序设计语言相关的类型长度而提供更高的编程创作能力和软件重用。脚本语言更适合在联系复杂的应用程序中进行胶着（粘合）。为了简化连接组件的工作，脚本语言被设计为无类型的，脚本语言一般是面向字符的，因为字符为许多不同的事物提供了一致的描述。

事实上，脚本语言都是动态语言，而动态语言都是解释型语言，不管它们是否是面向对象的语言。

149. A

解析1:

本题考查程序语言方面的基础知识。

编译是将高级语言源程序翻译成机器语言程序（汇编形式或机器代码形式），反编译是编译的逆过程。反编译通常不能把可执行文件还原成高级语言源代码，只能转换成功能上等价的汇编程序。

150. C

解析1:

本题考查程序语言基础知识。

函数调用时基本的参数传递方式有传值与传地址两种，在传值方式下是将实参的值传递给形参，因此实参可以是表达式（或常量），也可以是变量（或数组元素），这种信息传递是单方向的，形参不能再将值传回给实参。在传址方式下，需要将实参的地址传递给形参，因此，实参必须是变量（数组名或数组元素），不能是表达式（或常量）。这种方式下，被调用函数中对形式参数的修改实际上就是对实际参数的修改，因此客观上可以实现数据的双向传递。

151. A

解析1:

本题考查程序语言基础知识。

推导就是从文法的开始符号S出发，反复使用产生式，将产生式左部的非终结符替换为右部的文法符号序列（展开产生式用=>表示），直到产生一个终结符的序列时为止。从题中给出的分析树可得到如下的一个最左推导过程

$S \Rightarrow aAcB \Rightarrow aAaBcB \Rightarrow acaBcB \Rightarrow acabcB \Rightarrow acabcbScA \Rightarrow acabcbBdcA \Rightarrow acabcbdcA \Rightarrow acabcbdcc$ ，因此，acabcbdcc是该方法推导出的一个句子。其中用到的产生式如下：

$S \rightarrow aAcB \quad S \rightarrow Bd$

$A \rightarrow AaB \quad A \rightarrow c$

$B \rightarrow bScA \quad B \rightarrow b \quad B \rightarrow \varepsilon$

从起始符号S出发也可以如下推导： $S \Rightarrow Bd \Rightarrow \varepsilon d \Rightarrow d$ ，即该文法推导出的句子也可以为d，因此选项A错误。

152. B

解析1:

本题考查程序语言基础知识。

链表中的结点空间需要程序员根据需要申请和释放，因此，数据空间应采用堆存储分配策略。

153. C

解析1:

本题考查程序语言翻译基础知识。

词法分析是编译过程的第一阶段，其任务是对源程序从前到后（从左到右）逐个字符地扫描，从中识别出一个个的“单词”符号。语法分析的任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位，如“表达式”、“语句”和“程序”等。语义分析阶段主要检查源程序是否包含语义错误，并收集类型信息供后面的代码生成阶段使用。只有语法和语义都正确的源程序才能被翻译成正确的目标代码。目标代码生成是编译器工作的最后一个阶段。这一阶段的任务是把中间代码转换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码，这个阶段的工作与具体的机器密切相关。

源程序不可避免地会有一些错误，这些错误大致可分为语法错误和语义错误。语法错误是指语言结构上的使用错误，是指编译时所发现的程序错误，如单词拼写错误、标点符号错、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。

154. D

解析1:

本题考查程序语言基础知识。

“中间代码”是一种简单且含义明确的记号系统，与具体的机器无关，可以有若干种形式。可以将不同的高级程序语言翻译成同一种中间代码。由于与具体机器无关，使用中间代码有利于进行与机器无关的优化处理，以及提高编译程序的可移植性。

155. A

： 解析1：

本题考查程序语言基础知识。

变量是内存单元的抽象，用于在程序中表示数据。当变量存储的是内存单元地址时，称为指针变量，或者说指针变量指向了另一个变量。指针变量可以定义在函数或复合语句内，也可以定义在所有的函数之外，即可以是全局变量，也可以是局部变量。需要区分指针变量与指针所指向的变量，无论指针变量指向何种变量，其存储空间大小都是一样的。当指针变量指向数组中的一个元素时，对指针变量进行算术运算可以使其指向同一个数组中的其他元素。

156. C

解析1：

本题考查程序语言基础知识。

数据具有类型，便于编译程序在基础机器中完成对值的布局，同时还可用于检查表达式中对运算的应用是否正确。