

1.以下 Python 语言的模块中，（ ）不支持深度学习模型。

- A.TensorFlow B.Matplotlib C.PyTorch D.Keras

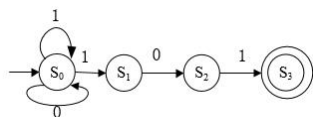
2.在 Python 语言中，（ ）是一种可变的、有序的序列结构，其中元素可以重复。

- A.元组(tuple) B.字符串(str) C.列表(list) D.集合(set)

3.Python 语言的特点不包括（ ）。

- A.跨平台、开源 B.编译型 C.支持面向对象程序设计 D.动态编程

4.下图所示为一个非确定有限自动机（NFA）， S_0 为初态， S_3 为终态。该 NFA 识别的字符串（ ）。

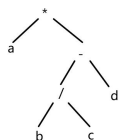


- A.不能包含连续的字符“0”
 B.不能包含连续的字符“1”
 C.必须以“101”开头
 D.必须以“101”结尾

5.对高级程序语言进行编译的过程中，使用（ ）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。

- A.决策表 B.符号表 C.广义表 D.索引表.

6.下图所示的二叉树表示的算术表达式是（ ）（其中的*、/、-表示乘、除、减运算）。



- A. $a*b/c-d$ B. $a*b/(c-d)$ C. $a*(b/c-d)$ D. $a*(b-c/d)$

7.函数 foo、hoo 的含义如下所示,函数调用 hoo(a,x)的两个参数分别采用引用调用(call by reference)和值调用(call by value)方式传递,则函数调用 foo(5)的输出结果为（ ）。

foo(int x)	hoo(int &x,int y)
<pre> int a=2; hoo(a,x); print(a,x);//输出a、x的值 </pre>	<pre> y=y+10; x=y-x. return; </pre>

- A.2, 5
 B.2, 15
 C.13, 5
 D.13, 15

8.设有描述简单算术表达的上下文无关文法如下，其中 **id** 表示单字母。

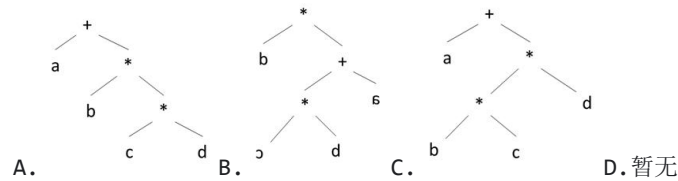
$E \rightarrow E+T \mid T$

$T \rightarrow F*T \mid F$

$F \rightarrow id$

与使用该文法描述的表达式 $a+b*c*d$ 相符的语法树为（ ），下图所示有限自动机（DFA）是（ ）。

问题 1:



问题 2:

- A. 确定的有限自动机，它能识别以 **bab** 结尾的
- B. 确定的有限自动机，他不能识别以 **bab** 结尾的
- C. 非确定的有限自动机，他能识别以 **bab** 结尾的
- D. 非确定的有限自动机，他不能识别以 **bab** 结尾的

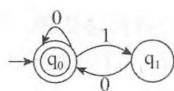
9.以编译方式翻译 C/C++ 源程序的过程中，类型检查在（ ）阶段处理

- A. 词法分析
- B. 语义分析
- C. 语法分析
- D. 目标代码生成

10.算数表达式 $a*(b+c/d)-e$ 的后缀式为（ ）。

- A. a b c d/+*e-
- B. a b c de*+/-
- C. a*b+c/d-e
- D. ab*cd/+e-

11.某有限自动机的状态转换图如下图所示，与该自动机等价的正规式是（ ）。



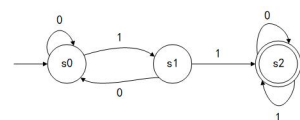
- A. $(0|1)^*$
- B. $(0|10)^*$
- C. $0^*(10)^*$
- D. $0^*(1|0)^*$

12.计算机执行程序时，内存分为静态数据区、代码区、栈区和堆区。其中（ ）一般在进行函数调用和返回时由系统进行控制和管理，（ ）由用户在程序中根据需要申请和释放。

问题 1: A. 静态数据区 B. 代码区 C. 栈区 D. 堆区

问题 2: A. 静态数据区 B. 代码区 C. 栈区 D. 堆区

13.某有限自动机的状态转换图如下图所示，该自动机可识别（ ）。



- A. 1001
- B. 1100
- C. 1010
- D. 0101

14.程序设计语言的大多数语法现象可以用 CFG (上下文无关文法)表示。下面的 CFG 产生式集用于描述简单算术表达式,其中+、-、*表示加、减、乘运算, id 表示单个字母表示的变量,那么符合该文法的表达式为 ()。

P: $E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow -F \mid id$

A. $a+-b-c$ B. $a*(b+c)$ C. $a*-b+2$ D. $-a/b+c$

15.函数 foo()、hoo()定义如下,调用函数 hoo()时,第一个参数采用传值(call by value)方式,第二个参数采用传引用(call by reference)方式。设有函数调(函数 foo(5)),那么"print(x)"执行后输出的值为 ()。

foo (int args)

```
int x=6;
hoo (args, x);
print (x);
```

hoo (int x, int& a)

```
x=x-1;
a=a*x;
return;
```

A.24 B.25 C.30 D.36

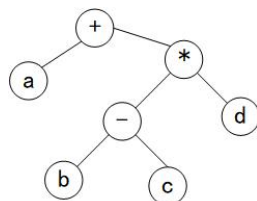
16.在程序的执行过程中,系统用 () 实现嵌套调用(递归调用)函数的正确返回。

A. 队列 B. 优先队列 C. 栈 D. 散列表

17.用 C/C++语言为某个应用编写的程序,经过 () 后形成可执行程序。

A. 预处理、编译、汇编、链接
B. 编译、预处理、汇编、链接
C. 汇编、预处理、链接、编译
D. 链接、预处理、编译、汇编

18.某表达式的语法树如下图所示,其后缀式(逆波兰式)是 ()。



A. $abcd-+*$ B. $ab-c+d*$ C. $abc-d*+$ D. $ab-cd+*$

19.表达式 $(a-b) * (c+d)$ 的后缀式(逆波兰式)是 ()。

A. $abcd-+*$ B. $ab-c+d*$ C. $abc-d*$ D. $ab-cd+*$

20.将编译器的工作过程划分为词法分析、语义分析、中间代码生成、代码优化和目标代码生成时,语法分析阶段的输入是 ()。若程序中的括号不配对,则会在 () 阶段检查出该错误。

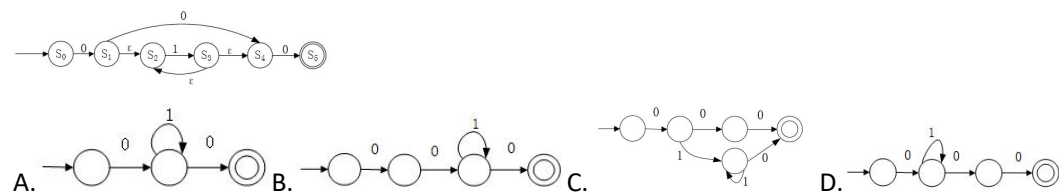
问题 1: A. 记号流 B. 字符流 C. 源程序 D. 分析树

问题 2: A. 词法分析 B. 语法分析 C. 语义分析 D. 目标代码生成

21.函数 f、g 的定义如下,执行表达式“ $y = f(2)$ ”的运算时,函数调用 g(1a)分别采用引用调用(call by reference)方式和值调用(call by value)方式,则该表达式求值结束后 y 的值分别为 ()。

A. 9、6 B. 20、6 C. 20、9 D. 30、9

22. 下图所示为一个不确定有限自动机 (NFA) 的状态转换图, 与该 NFA 等价的 DFA 是 ()。



23. 在以阶段划分的编译器中, () 阶段的主要作用是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。

- A. 词法分析 B. 语法分析 C. 语义分析 D. 代码生成

24. 在对高级语言源程序进行编译或解释处理的过程中, 需要不断收集、记录和使用源程序中一些相关符号的类型和特征等信息, 并将其存入 () 中。

- A. 哈希表 B. 符号表 C. 堆栈 D. 队列

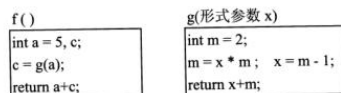
25. 以编译方式翻译 C/C++ 源程序的过程中, () 阶段的主要任务是对各条语句的结构进行合法性分析。

- A. 词法分析 B. 语义分析 C. 语法分析 D. 目标代码生成

26. 通用的高级程序设计语言一般都会提供描述数据、运算、控制和数据传输的语言成分, 其中, 控制包括顺序、() 和循环结构。

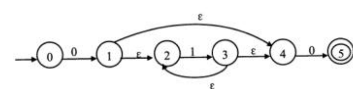
- A. 选择 B. 递归 C. 递推 D. 函数

27. 函数 f 和 g 的定义如下图所示。执行函数 f 时若采用引用 (call by reference) 方式调用函数 $g(a)$, 则函数 f 的返回值为 ()。



- A. 14 B. 18 C. 24 D. 28

28. 下图所示为一个不确定有限自动机 (NFA) 的状态转换图。该 NFA 可识别字符串 ()。



- A. 0110 B. 0101 C. 1100 D. 1010

29. 在以阶段划分的编译器中, () 阶段的主要作用是分析程序中的句子结构是否正确。

- A. 词法分析 B. 语法分析 C. 语义分析 D. 代码生成

30. 在程序运行过程中, () 时涉及整型数据转换为浮点型数据的操作。

- A. 将浮点型变量赋值给整型变量
B. 将整型常量赋值给整型变量
C. 将整型变量与浮点型变量相加
D. 将浮点型常量与浮点型变量相加

31.下面二叉树表示的简单算术表达式为（ ）。

- A. $10*20+30-40$ B. $10*(20+30-40)$ C. $10*(20+30)-40$ D. $10*20+(30-40)$

32.语法指导翻译是一种（ ）方法。

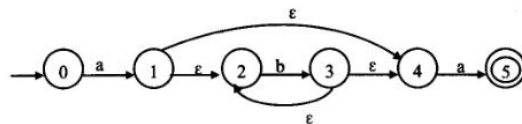
- A. 动态语义分析 B. 中间代码优化 C. 静态语义分析 D. 目标代码优化

33.简单算术表达式的结构可以用下面的上下文无关文法进行描述（E 为开始符号），（ ）是符合该文法的句子。

$E \rightarrow T \mid E+T$
 $T \rightarrow F \mid T * F$
 $F \rightarrow -F \mid N$
 $N \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

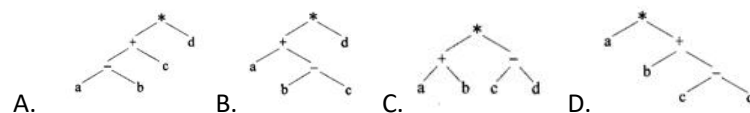
- A. $2--3*4$ B. $2+-3*4$ C. $(2+3)*4$ D. $2*4-3$

34.下图所示为一个不确定有限自动机（NFA）的状态转换图。该 NFA 识别的字符串集合可用正规式（ ）描述。



- A. ab^*a B. $(ab)^*a$ C. a^*ba D. $a(ba)^*$

35.对于后缀表达式 $a \ b \ c \ - \ + \ d \ *$ （其中， $-$ 、 $+$ 、 $*$ 表示二元算术运算减、加、乘），与该后缀式等价的语法树为（ ）。



36.将高级语言源程序翻译为可在计算机上执行的形式有多种不同的方式，其中（ ）。

- A. 编译方式和解释方式都生成逻辑上与源程序等价的目标程序
 B. 编译方式和解释方式都不生成逻辑上与源程序等价的目标程序
 C. 编译方式生成逻辑上与源程序等价的目标程序，解释方式不生成
 D. 解释方式生成逻辑上与源程序等价的目标程序，编译方式不生成

37.（ ）是一种函数式编程语言。

- A. Lisp B. Prolog C. Python D. Java/C++

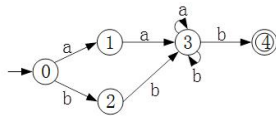
38.函数 `main()`、`f()` 的定义如下所示。调用函数 `f()` 时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，则函数 `main()` 执行后输出的值为（ ）。

```
main()
{
    int x = 10;
    f(x, x);
    print(x);
}
```

```
f(int x, int &a)
{
    x = 2*x - 1;
    a = a + x;
    return;
}
```

- A. 10 B. 19 C. 20 D. 29

39.某确定的有限自动机（DFA）的状态转换图如下图所示（0 是初态，4 是终态），则该 DFA 能识别（ ）。



- A. aaab B. abab C. bbba D. abba

40.编译过程中进行的语法分析主要是分析（ ）。

- A. 源程序中的标识符是否合法 B. 程序语句的含义是否合法
C. 程序语句的结构是否合法 D. 表达式的类型是否合法

41.将高级语言源程序通过编译或解释方式进行翻译时，可以先生成与源程序等价的某种中间代码。以下关于中间代码的叙述中，正确的是（ ）。

- A. 中间代码常采用符号表来表示
B. 后缀式和三地址码是常用的中间代码
C. 对中间代码进行优化要依据运行程序的机器特性
D. 中间代码不能跨平台

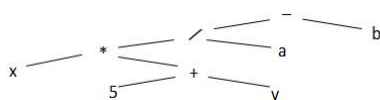
42.以下关于程序设计语言的叙述中，不正确的是（ ）。

- A. 脚本语言中不使用变量和函数
B. 标记语言常用于描述格式化和链接
C. 脚本语言采用解释方式实现
D. 编译型语言的执行效率更高

43.更适合用来开发操作系统的编程语言是（ ）。

- A. C/C++ B. Java C. Python D. JavaScript

44.下图为一个表达式的语法树，该表达式的后缀形式为（ ）。



- A. x 5 y + * a / b -
B. x 5 y a b * + / -
C. - / * + x 5 y a b
D. x 5 + * y + a / b -

45.函数 main()、f() 的定义如下所示，调用函数们 f() 时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，main() 函数中 “print(x)” 执行后输出的值为（ ）。

main()

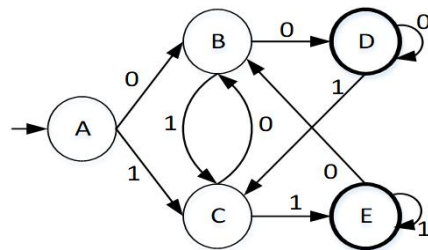
```
int x = 5;
f(x+1, x);
print(x);
```

f(int x, int &a)

```
x = x * x - 1;
a = x + a;
return;
```

- A. 11 B. 40 C. 45 D. 70

46.某确定的有限自动机（DFA）的状态转换图如下图所示（A 是初态，D、E 是终态），则该 DFA 能识别（ ）。



- A.00110 B.10101 C.11100 D.11001

47.在以阶段划分的编译过程中，判断程序语句的形式是否正确属于（ ）阶段的工作。

- A.词法分析 B.语法分析 C.语义分析 D.代码生成

48.在仅由字符 a、b 构成的所有字符串中，其中以 b 结尾的字符串集合可用正规式表示为（ ）。

- A. $(b|ab)^*b$ B. $(ab)^*b$ C. a^*b^*b D. $(a|b)^*b$

49.在高级语言源程序中，常需要用户定义的标识符为程序中的对象命名，常见的命名对象有（ ）。

①关键字（或保留字）②变量③函数④数据类型⑤注释

- A.①②③ B.②③④ C.①③⑤ D.②④⑤

50.运行下面的 C 程序代码段，会出现（ ）错误。

```

int k=0;
for(;k<100;);
{k++;}
  
```

- A.变量未定义 B.静态语义 C.语法 D.动态语义

51.乔姆斯基（Chomsky）将文法分为 4 种类型，程序设计语言的大多数语法现象可用其中的（ ）描述。

- A.上下文有关文法 B.上下文无关文法 C.正规文法 D.短语结构文法

52.由字符 a、b 构成的字符串中，若每个 a 后至少跟一个 b，则该字符串集合可用正规式表示为（ ）。

- A. $(b|ab)^*$ B. $(ab)^*$ C. $(a^*b^*)^*$ D. $(a|b)^*$

53.常用的函数参数传递方式有传值与传引用两种。（ ）。

- A.在传值方式下，形参与实参之间互相传值
 B.在传值方式下，实参不能是变量
 C.在传引用方式下，修改形参实质上改变了实参的值
 D.在传引用方式下，实参可以是任意的变量和表达式

54.以下关于高级程序设计语言实现的编译和解释方式的叙述中，正确的是（ ）。

- A.编译程序不参与用户程序的运行控制，而解释程序则参与
 B.编译程序可以用高级语言编写，而解释程序只能用汇编语言编写
 C.编译方式处理源程序时不进行优化，而解释方式则进行优化
 D.编译方式不生成源程序的目标程序，而解释方式则生成

55. 以下关于脚本语言的叙述中，正确的是（ ）。

- A. 脚本语言是通用的程序设计语言
- B. 脚本语言更适合应用在系统级程序开发中
- C. 脚本语言主要采用解释方式实现
- D. 脚本语言中不能定义函数和调用函数

56. 将高级语言源程序先转化为一种中间代码是现代编译器的常见处理方式。常用的中间代码有后缀式、（ ）、树等。

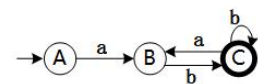
- A. 前缀码
- B. 三地址码
- C. 符号表
- D. 补码和移码

57. 函数 `main()`、`f()` 的定义如下所示，调用函数 `f()` 时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，`main` 函数中“`print(x)`”执行后输出的值为（ ）。

<code>main()</code>	<code>f(int x, int &a)</code>
<pre>int x=1; f(5, x); print(x);</pre>	<pre>x=2*x+1; a=a+x; return;</pre>

- A. 1
- B. 6
- C. 11
- D. 12

58. 某确定的有限自动机（DFA）的状态转换图如下图所示（A 是初态，C 是终态），则该 DFA 能识别（ ）。



- A. aabb
- B. abab
- C. baba
- D. abba

59. 移进-归约分析法是编译程序（或解释程序）对高级语言源程序进行语法分析的一种方法，属于（ ）的语法分析方法。

- A. 自顶向下（或自上而下）
- B. 自底向上（或自下而上）
- C. 自左向右
- D. 自右向左

60. 函数 `t()`、`f()` 的定义如下所示，若调用函数 `t` 时传递给 `x` 的值为 5，并且调用函数 `F()` 时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，则函数 `t` 的返回值为（ ）。

<code>t(int x)</code>	<code>f(int r, int &s)</code>
<pre>int a; a = 3*x+1; f(x, a); return a-x;</pre>	<pre>int x; x = 2*s+1; s = x+r; r = x-1; return;</pre>

- A. 33
- B. 22
- C. 11
- D. 负数

61. 某非确定的有限自动机（NFA）的状态转换图如下图所示（`q0` 既是初态也是终态）。以下关于该 NFA 的叙述中，正确的是（ ）。

- A. 其可识别的 0、1 序列的长度为偶数
- B. 其可识别的 0、1 序列中 0 与 1 的个数相同
- C. 其可识别的非空 0、1 序列中开头和结尾字符都是 0
- D. 其可识别的非空 0、1 序列中结尾字符是 1

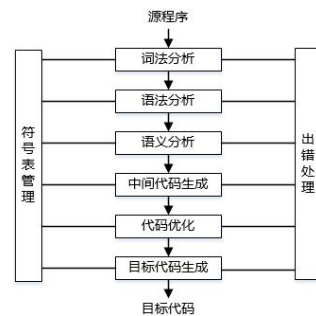
62. 某程序运行时陷入死循环，则可能的原因是程序中存在（ ）。

- A. 词法错误
- B. 语法错误
- C. 动态的语义错误
- D. 静态的语义错误

63.表达式采用逆波兰式表示时，利用（ ）进行求值。

- A. 栈 B. 队列 C. 符号表 D. 散列表

64.编译器和解释器是两种基本的高级语言处理程序。编译器对高级语言源程序的处理过程可以划分为词法分析、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，其中，（ ）并不是每个编译器都必需的，与编译器相比，解释器（ ）。



问题 1:

- A. 词法分析和语法分析 B. 语义分析和中间代码生成
C. 中间代码生成和代码优化 D. 代码优化和目标代码生成

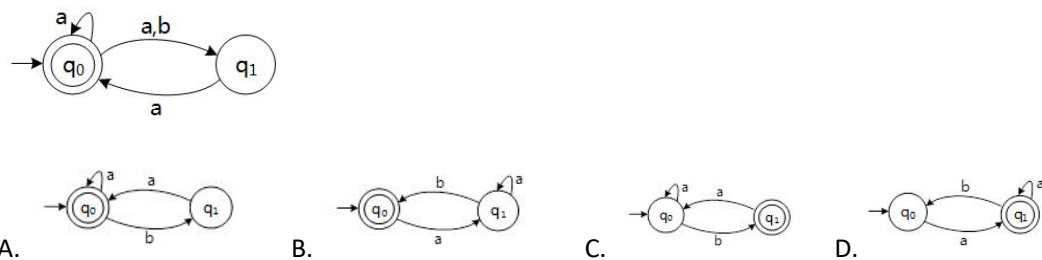
问题 2:

- A. 不参与运行控制，程序执行的速度慢 B. 参与运行控制，程序执行的速度慢
C. 参与运行控制，程序执行的速度快 D. 不参与运行控制，程序执行的速度快

65.递归下降分析方法是一种（ ）方法。

- A. 自底向上的语法分析 B. 自上而下的语法分析
C. 自底向上的词法分析 D. 自上而下的词法分析

66.某非确定的有限自动机（NFA）的状态转换图如下图所示（ q_0 既是初态也是终态），与该 NFA 等价的确定的有限自动机（DFA）是



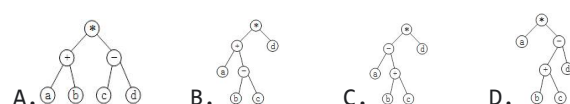
67.对高级语言源程序进行编译或解释的过程可以分为多个阶段，解释方式不包含（ ）阶段。

- A. 词法分析 B. 语法分析 C. 语义分析 D. 目标代码生成

68.C 程序中全局变量的存储空间在（ ）分配。

- A. 代码区 B. 静态数据区 C. 栈区 D. 堆区

69.与算术表达式“(a+(b-c))*d”对应的树是（ ）。



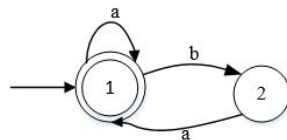
70. 以下关于程序设计语言的叙述中，错误的是（ ）。

- A. 程序设计语言的基本成分包括数据、运算、控制和传输等
- B. 高级程序设计语言不依赖于具体的机器硬件
- C. 程序中局部变量的值在运行时不能改变
- D. 程序中常量的值在运行时不能改变

71. 对于大多数通用程序设计语言，用（ ）描述其语法即可。

- A. 正规文法
- B. 上下文无关文法
- C. 上下文有关文法
- D. 短语结构文法

72. 以下关于下图所示有限自动机的叙述中，不正确的是（ ）。



- A. 该自动机识别的字符串中 a 不能连续出现
- B. 自动机识别的字符串中 b 不能连续出现
- C. 该自动机识别的非空字符串必须以 a 结尾
- D. 该自动机识别的字符串可以为空串

73. 对高级语言源程序进行编译的过程可以分为多个阶段，分配寄存器的工作在（ ）阶段进行。

- A. 词法分析
- B. 语法分析
- C. 语义分析
- D. 目标代码生成

74. 将高级语言源程序翻译成机器语言程序的过程中，常引入中间代码。以下关于中间代码的叙述中，不正确的是（ ）。

- A. 中间代码不依赖于具体的机器
- B. 使用中间代码可提高编译程序的可移植性
- C. 中间代码可以用树或图表示
- D. 中间代码可以用栈和队列表示

75. 算术表达式“(a-b)*(c+d)”的后缀式是（ ）。“”

- A. ab-cd+ *
- B. abcd-*+
- C. ab-+*cd
- D. ab-c+d*

76. 属于面向对象、解释型程序设计语言的是（ ）。

- A. XML
- B. Python
- C. Prolog
- D. C

77. 在 C/C++ 程序中，整型变量 a 的值为 0 且应用在表达式“c=b/a”中，则最可能发生的情形是（ ）。

- A. 编译时报告有语法错误
- B. 编译时报告有逻辑错误
- C. 运行时报告有语法错误
- D. 运行时产生异常

78. 大多数程序设计语言的语法规则用（ ）描述即可。

- A. 正规文法
- B. 上下文无关文法
- C. 上下文有关文法
- D. 短语结构文法

79.以下关于实现高级程序设计语言的编译和解释方式的叙述中，正确的是（ ）。

- A.在编译方式下产生源程序的目标程序，在解释方式下不产生
- B.在解释方式下产生源程序的目标程序，在编译方式下不产生
- C.编译和解释方式都产生源程序的目标程序，差别是优化效率不同
- D.编译和解释方式都不产生源程序的目标程序，差别在是否优化

80.编译程序对高级语言源程序进行编译的过程中，要不断收集、记录和使用源程序中一些相关符号的类型和特征等信息，并将其存入（ ）中。

- A.符号表
- B.哈希表
- C.动态查找表
- D.栈和队列

81.引用调用方式下进行函数调用，是将（ ）。

- A.实参的值传递给形参
- B.实参的地址传递给形参
- C.形参的值传递给实参
- D.形参的地址传递给实参

82.以下程序设计语言中，（ ）更适合用来进行动态网页处理。

- A.HTML
- B.LISP
- C.PHP
- D.JAVA/C++

83.弱类型的语言（动态类型语言）是指不需要进行变量/对象类型声明的语言。（ ）属于弱类型语言。

- A.Java
- B.C/C++
- C.Python
- D.C#

84.对高级语言源程序进行编译的过程中，有穷自动机（NFA 或 DFA）是进行（ ）的适当工具。

- A.词法分析
- B.语法分析
- C.语义分析
- D.出错处理

85.将高级语言程序翻译为机器语言程序的过程中，常引入中间代码，其好处是（ ）。

- A.有利于进行反编译处理
- B.有利于进行与机器无关的优化处理
- C.尽早发现语法错误
- D.可以简化语法和语义分析

86.算术表达式 $a + (b - c) * d$ 的后缀式是（ ）（-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

- A. $bc - d * a +$
- B. $abc - d * +$
- C. $ab + c - d *$
- D. $abcd - * +$

87.已知文法 $G: S \rightarrow A0 | B1, A \rightarrow S1 | 1, B \rightarrow S0 | 0$, 其中 S 是开始符号。从 S 出发可以推导出（ ）。

- A.所有由 0 构成的字符串
- B.所有由 1 构成的字符串
- C.某些 0 和 1 个数相等的字符串
- D.所有 0 和 1 个数不同的字符串

88. 程序运行过程中常使用参数在函数（过程）间传递消息，引用调用传递的是实参的（ ）。

- A.地址
- B.类型
- C.名称
- D.值

89.编译过程中，对高级语言程序语名的翻译主要考虑声明语名和可执行语句。对声明语句，主要是将所需要的信息正确地填入合理组织的（ ）中；对可执行语句，则是（ ）。

问题 1: A.符号表 B.栈 C.队列 D.树

问题 2:

- A.翻译成机器代码并加以执行
- B.转换成语法树
- C.翻译成中间代码或目标代码
- D.转换成有限自动机

90. 以下关于语言 $L=\{a^n b^n \mid n \geq 1\}$ 的叙述中，正确的是（ ）。

- A. 可用正规式“ $aa*bb*$ ”描述，但不能通过有限自动机识别
- B. 可用正规式“ $a^m b^m$ ”表示，但可用有限自动机识别
- C. 不能用正规式表示，但可用有限自动机识别
- D. 既不能用正规式表示，也不能通过有限自动机识别

91. 在对高级语言源程序进行编译的过程中，为源程序中变量所分配的存储单元的地址属于（ ）。

- A. 逻辑地址
- B. 物理地址
- C. 接口地址
- D. 线性地址

92. 以下关于传值调用与引用调用的叙述中，正确的是（ ）。

- ①在传值调用方式下，可以实现形参和实参间双向传递数据的效果
- ②在传值调用方式下，实参可以是变量，也可以是常量和表达式
- ③在引用调用方式下，可以实现形参和实参间双向传递数据的效果
- ④在引用调用方式下，实参可以是变量，也可以是常量和表达式

- A. ①③
- B. ①④
- C. ②③
- D. ②④

93. 以下关于解释程序和编译程序的叙述中，正确的是（ ）。

- A. 编译程序和解释程序都生成源程序的目标程序
- B. 编译程序和解释程序都不生成源程序的目标程序
- C. 编译程序生成源程序的目标程序，而解释程序则不然
- D. 编译程序不生成源程序的目标程序，而解释程序反之

94. 以下关于程序错误的叙述中，正确的是（ ）。

- A. 编译正确的程序必然不包含语法错误
- B. 编译正确的程序必然不包含语义错误
- C. 除数为 0 的错误可以在语义分析阶段检查出来
- D. 除数为 0 的错误可以在语法分析阶段检查出来

95. 在对程序语言进行翻译的过程中，常采用一些与之等价的中间代码表示形式。常用的中间代码表示不包括（ ）。

- A. 树
- B. 后缀式
- C. 四元式
- D. 正则式

96. 将高级语言源程序翻译成目标程序的是（ ）。

- A. 解释程序
- B. 编译程序
- C. 链接程序
- D. 汇编程序

97. 算术表达式 $(a-b)*c+d$ 的后缀式是（ ）（-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

- A. $a\ b\ c\ d\ -\ *\ +$
- B. $a\ b\ -\ c\ d\ *\ +$
- C. $a\ b\ -\ c\ *\ d\ +$
- D. $a\ b\ c\ -\ d\ *\ +$

98. 语言 $L=\{a^m b^n \mid m \geq 0, n \geq 1\}$ 的正规表达式是（ ）。

- A. $aa*bb*$
- B. $a*bb*$
- C. $aa*b*$
- D. $a*b*$

99. 可用于编写独立程序和快速脚本的语言是（ ）。

- A. Python
- B. Prolog
- C. Java
- D. C#

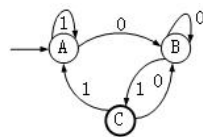
100.编译和解释是实现高级程序设计语言翻译的两种基本形式。以下关于编译与解释的叙述中，正确的是（ ）。

- A.在解释方式下，对源程序不进行词法分析和语法分析，直接进行语义分析
- B.在解释方式下，无需进行词法、语法和语义分析，而是直接产生源程序的目标代码
- C.在编译方式下，必须进行词法、语法和语义分析，然后再产生源程序的目标代码
- D.在编译方式下，必须先形成源程序的中间代码，然后再产生与机器对应的目标代码

101.函数（过程）调用时，常采用传值与传地址两种方式在实参与形参间传递信息。以下叙述中，正确的是（ ）。

- A.在传值方式下，将形参的值传给实参，因此，形参必须是常量或变量
- B.在传值方式下，将实参的值传给形参，因此，实参必须是常量或变量
- C.在传地址方式下，将形参的地址传给实参，因此，形参必须有地址
- D.在传地址方式下，将实参的地址传给形参，因此，实参必须有地址

102.下图所示为一个有限自动机（其中，A 是初态、C 是终态），该自动机所识别的字符串的特点是（ ）。



- A.必须以 11 结尾的 0、1 串
- B.必须以 00 结尾的 0、1 串
- C.必须以 01 结尾的 0、1 串
- D.必须以 10 结尾的 0、1 串

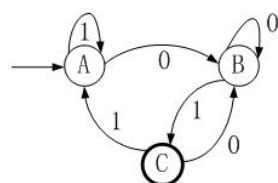
103.算术表达式 $x - (y + c) * 8$ 的后缀式是（ ）（-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

- | | |
|--------------------------------|--------------------------------|
| A. $x \ y \ c \ 8 \ - \ + \ *$ | B. $x \ y \ - \ c \ + \ 8 \ *$ |
| C. $x \ y \ c \ 8 \ * \ - \ +$ | D. $x \ y \ c \ + \ 8 \ * \ -$ |

104.函数 t、f 的定义如下所示，其中，a 是整型全局变量。设调用函数 t 前 a 的值为 5，则在函数 t 中以传值调用(call by value)方式调用函数 f 时，输出为（ ）；在函数 t 中以引用调用(call by reference)方式调用函数 f 时，输出为（ ）。

- 问题 1: A.12 B.16 C.20 D.24
- 问题 2: A.12 B.16 C.20 D.24

105.下图所示为一个有限自动机（其中，A 是初态、C 是终态），该自动机识别的语言可用正规式（ ）表示。



- | | | | |
|----------------|------------------|-----------------|---------------------|
| A. $(0 1)^*01$ | B. $1^*0^*10^*1$ | C. $1^*(0)^*01$ | D. $1^*(0 10)^*1^*$ |
|----------------|------------------|-----------------|---------------------|

106.某程序设计语言规定在源程序中的数据都必须具有类型，然而，（ ）并不是做出此规定的理由。

- A. 为数据合理分配存储单元
- B. 可以定义和使用动态数据结构
- C. 可以规定数据对象的取值范围及能够进行的运算
- D. 对参与表达式求值的数据对象可以进行合法性检查

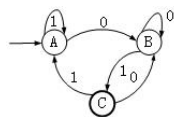
107.若 C 程序的表达式中引用了未赋初值的变量，则（ ）。

- A. 编译时一定会报告错误信息，该程序不能运行
- B. 可以通过编译并运行，但运行时一定会报告异常
- C. 可以通过编译，但链接时一定会报告错误信息而不能运行
- D. 可以通过编译并运行，但运行结果不一定是期望的结果

108.传值与传地址是函数调用时常采用的信息传递方式，（ ）。

- A. 在传值方式下，是将形参的值传给实参
- B. 在传值方式下，形参可以是任意形式的表达式
- C. 在传地址方式下，是将实参的地址传给形参
- D. 在传地址方式下，实参可以是任意形式的表达式

109.下图所示为一个有限自动机（其中，A 是初态、C 是终态），该自动机可识别（ ）。



- A. 0000
- B. 1111
- C. 0101
- D. 1010

110.以下关于高级程序设计语言翻译的叙述中，正确的是（ ）。

- A. 可以先进行语法分析，再进行词法分析
- B. 在语法分析阶段可以发现程序中的所有错误
- C. 语义分析阶段的工作与目标机器的体系结构密切相关
- D. 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.若一种程序设计语言规定其程序中的数据必须具有类型，则有利于（ ）。

- ①在翻译程序的过程中为数据合理分配存储单元
- ②对参与表达式计算的数据对象进行检查
- ③定义和应用动态数据结构
- ④规定数据对象的取值范围及能够进行的运算
- ⑤对数据进行强制类型转换

- A. ①②③
- B. ①②④
- C. ②④⑤
- D. ③④⑤

112.算术表达式采用逆波兰式表示时不用括号，可以利用（ ）进行求值。与逆波兰式 $ab-cd+*$ 对应的中缀表达式是（ ）。

问题 1: A. 数组 B. 栈 C. 队列 D. 散列表

问题 2: A. $a-b+c*d$ B. $(a-b)*c+d$ C. $(a-b)*(c+d)$ D. $a-b*c+d$

113.下面 C 程序段中 `count++` 语句执行的次数为 ()。

```
for(int i=1; i<=11; i*=2)
    for(int j=1; j<=i; j++)
        count++;
```

A.15 B.16 C.31 D.32

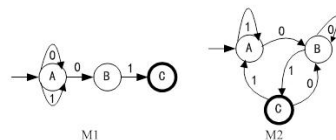
114.以下关于汇编语言的叙述中，错误的是 ()。

- A. 汇编语言源程序中的指令语句将被翻译成机器代码
- B. 汇编程序先将源程序中的伪指令翻译成机器代码，然后再翻译指令语句
- C. 汇编程序以汇编语言源程序为输入，以机器语言表示的目标程序为输出
- D. 汇编语言的指令语句必须具有操作码字段，可以没有操作数字段

115.以下关于可视化程序设计的叙述中，错误的是 ()。

- A. 可视化程序设计使开发应用程序无需编写程序代码
- B. 可视化程序设计基于面向对象的思想，引入了控件和事件驱动
- C. 在可视化程序设计中，构造应用程序界面就像搭积木
- D. 在可视化程序设计中，采用解释方式可随时查看程序的运行效果

116.下图所示为两个有限自动机 M1 和 M2 (A 是初态、C 是终态)，()。



- A. M1 和 M2 都是确定的有限自动机
- B. M1 和 M2 都是不确定的有限自动机
- C. M1 是确定的有限自动机，M2 是不确定的有限自动机
- D. M1 是不确定的有限自动机，M2 是确定的有限自动机

117.下图所示的有限自动机中，0 是初始状态，3 是终止状态，该自动机可以识别 ()。

- A. abab B. aaaa
- C. bbbb D. abba

118.编译程序分析源程序的阶段依次是 ()。

- A. 词法分析、语法分析、语义分析
- B. 语法分析、词法分析、语义分析
- C. 语义分析、语法分析、词法分析
- D. 语义分析、词法分析、语法分析

119.以下关于变量和常量和叙述中，错误的是 ()。

- A. 变量的取值在程序运行过程中可以改变，常量则不行
- B. 变量具有类型属性，常量则没有
- C. 变量具有对应的存储单元，常量则没有
- D. 可以对变量赋值，不能对常量赋值

120.对于正规式 $0^*(10^*1)^*0^*$ ，其正规集中字符串的特点是（ ）。

- A.开头和结尾必须是 0 B.1 必须出现偶数次 C.0 不能连续出现 D.1 不能连续出现

121.标记语言用一系列约定好的标记来对电子文档进行标记，以实现对其语义、结构及格式的定义。（ ）不是标记语言。

- A.HTML B.XML C.WML D.PHP

122.以下关于高级语言程序的编译和解释的叙述中，正确的是（ ）。

- A.编译方式下，可以省略对源程序的词法分析、语法分析
B.解释方式下，可以省略对源程序的词法分析、语法分析
C.编译方式下，在机器上运行的目标程序完全独立于源程序
D.解释方式下，在机器上运行的目标程序完全独立于源程序

123.程序的三种基本控制结构是（ ）。

- A.过程、子程序分程序 B.顺序、选择和重复
C.递归、堆栈和队列 D.调用、返回和跳转

124.编译程序对 C 语言源程序进行语法分析时，可以确定（ ）。

- A.变量是否定义（或声明） B.变量的值是否正确
C.循环语句的执行次数 D.循环条件是否正确

125.逻辑表达式“ $a \wedge b \vee c \wedge (b \vee x > 0)$ ”的后缀式为（ ）。（其中 \wedge 、 \vee 分别表示逻辑与、逻辑或， $>$ 表示关系运算大于，对逻辑表达式进行短路求值）

- A. $abcbx0 > \vee \wedge \wedge \vee$
B. $ab \wedge c \vee b \wedge x0 > \vee$
C. $ab \wedge cb \wedge x > 0 \vee \vee$
D. $ab \wedge cbx0 > \vee \wedge \vee$

126.下面的 C 程序代码段在运行中会出现（ ）错误。

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

- A.语法 B.类型不匹配 C.变量定义 D.动态语义

127.与逆波兰式 $ab+ -c*d-$ 对应的中缀表达式是（ ）。

- A. $a-b-c*d$ B. $-(a+b)*c-d$ C. $-a+b*c-d$ D. $(a+b)*(-c-d)$

128.对于下面的文法 $G[S]$ ，（ ）是其句子（从 S 出发开始推导）。

$G[S]: S \rightarrow M | (S, M) \quad M \rightarrow P | MP \quad P \rightarrow a | b | c | \dots | x | x | z$

- A. $((a, F))$
B. $((fac, bb), g)$
C. (abc)
D. $(c, (da))$

129.高级程序设计语言中用于描述程序中的运算步骤、控制结构及数据传输的是（ ）。

- A. 语句 B. 语义 C. 语用 D. 语法

130.开发专家系统时，通过描述事实和规则由模式匹配得出结论，这种情况下适用的开发语言是（ ）。

- A.面向对象语言 B.函数式语言 C.过程式语言 D.逻辑式语言

131.程序设计语言中（ ）。

- A.while 循环语句的执行效率比 do-while 循环语句的执行效率高
B.while 循环语句的循环体执行次数比循环条件的判断次数多 1，而 do-while 语句的循环体执行次数比循环条件的判断次数少 1
C.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比循环条件的判断次数多 1
D.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.函数 t()、f()的定义如下所示，若调用函数 t 时传递给 x 的值为 3，并且调用函数 f()时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，则函数 t 的返回值为（ ）。

<pre>t(int x) { int a; a = 3 * x - 1; f(x, a); return a + x; }</pre>	<pre>f(int r, int s) { int x; x = 2 * r + 1; s = x * r; r = s - x; return; }</pre>
--	--

- A.35 B.24 C.22 D.11

133.下面关于编程语言的各种说法中，（ ）是正确的。

- A.由于 C 语言程序是由函数构成的，因此它是一种函数型语言
B.Smalltalk、C++、Java、C#都是面向对象语言
C.函数型语言适用于编写处理高速计算的程序，常用于超级计算机的模拟计算
D.逻辑型语言是在 Client/Server 系统中用于实现负载分散的程序语言

134.设某上下文无关文法如下： $S \rightarrow 11 \mid 1001 \mid S0 \mid SS$ ，则该文法所产生的所有二进制字符串都具有的特点是（ ）。

- A.能被 3 整除 B.0、1 出现的次数相等 C.0 和 1 的出现次数都为偶数 D.能被 2 整除

135.高级语言源程序的编译过程分若干个阶段，分配寄存器属于（ ）阶段的工作。

- A.词法分析 B.语法分析 C.语义分析 D.代码生成

136.给定文法 $G[S]$ 及其非终结符 A， $FIRST(A)$ 定义为：从 A 出发能推导出的终结符号的集合（S 是文法的起始符号，为非终结符）。对于文法 $G[S]$ ：

$S \rightarrow [L] \mid a$

$L \rightarrow L, S \mid S$

其中， $G[S]$ 包含的四个终结符号分别为：

a, [,]

则 $FIRST(S)$ 的成员包括（ ）。

- A.a B.[C.[和] D.a、[、]和,

137.表达式 $(a-b)*(c+5)$ 的后缀式是 ()。

- A. $a \ b \ c+*-$ B. $a \ b-c+5*$ C. $a \ b \ c-*5+$ D. $a \ b \ -c \ 5+*$

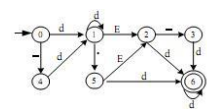
138.程序设计语言一般都提供多种循环语句，例如实现先判断循环条件再执行循环体的 **while** 语句和先执行循环体再判断循环条件的 **do-while** 语句。关于这两种循环语句，在不改变循环体的条件下，()是正确的。

- A. **while** 语句的功能可由 **do-while** 语句实现
 B. **do-while** 语句的功能可由 **while** 语句实现
 C. 若已知循环体的次数，则只能使用 **while** 语句
 D. 循环条件相同时，**do-while** 语句的执行效率更高

139.编译程序对高级语言源程序进行翻译时，需要在该程序的地址空间中为变量指定地址，这种地址称为 ()。

- A. 逻辑地址 B. 物理地址 C. 接口地址 D. 线性地址

140.某确定性有限自动机 (DFA) 的状态转换图如下图所示，令 $d=0|1|2|\dots|9$ ，则以下字符串中，能被该 DFA 接受的是 ()。



- A. 3857 B. 1.2E+5 C. -123.67 D. 0.576E10

141.有限自动机 (FA) 可用于识别高级语言源程序中的记号 (单词)，FA 可分为确定的有限自动机 (DFA) 和不确定的有限自动机 (NFA)。若某 DFA D 与某 NFA M 等价，则 ()。

- A. DFA D 与 NFA M 的状态数一定相等
 B. DFA D 与 NFA M 可识别的记号相同
 C. NFA M 能识别的正规集是 DFA D 所识别正规集的真子集
 D. DFA D 能识别的正规集是 NFA M 所识别正规集的真子集

142.下列叙述中错误的是 ()。

- A. 面向对象程序设计语言可支持过程化的程序设计
 B. 给定算法的时间复杂性与实现该算法所采用的程序设计语言无关
 C. 与汇编语言相比，采用脚本语言编程可获得更高的运行效率
 D. 面向对象程序设计语言不支持对一个对象的成员变量进行直接访问

143.已知某文法 $G[S]: S \rightarrow 0S0 \mid S \rightarrow 1$ ，从 S 推导出的符号串可用 () ($n \geq 0$) 描述。

- A. $(010)^n$ B. $0^n 1 0^n$ C. 1^n D. $01^n 0$

144.编译器对高级语言源程序的处理过程可以划分为词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成等几个阶段，其中，()并不是每种编译器都必需的。

- A. 词法分析和语法分析
 B. 语义分析和中间代码生成
 C. 中间代码生成和代码优化
 D. 代码优化和目标代码生成

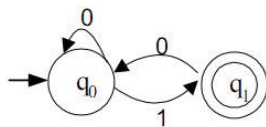
145. 设某语言的语法规则用上下文无关文法 $G=(N, T, P, S)$ 表示, 其中 N 是非终结符号的集合, T 是终结符号的集合, P 是产生式集合, S 是开始符号, 令 $V=N \cup T$, 那么符合该语言的句子是 ()。

- A. 从 S 出发推导的、仅包含 T 中符号的符号串
- B. 从 N 中符号出发推导的、仅包含 T 中符号的符号串
- C. 从 S 出发推导的、包含 V 中符号的符号串
- D. 从 N 中符号出发推导的、包含 V 中符号的符号串

146. 由 a 、 b 构造且仅包含偶数个 a 的串的集合用正规式表示为 ()。

- A. $(a^*a)^*b^*$
- B. $(b^*(ab^*a)^*)^*$
- C. $(a^*(ba^*)^*b)^*$
- D. $(a|b)^*(aa)^*$

147. 下图所示有限自动机的特点是 ()。



- A. 识别的 0 、 1 串是以 0 开头且以 1 结尾
- B. 识别的 0 、 1 串中 1 的数目为偶数
- C. 识别的 0 、 1 串中 0 后面必须是 1
- D. 识别的 0 、 1 串中 1 不能连续出现

148. 下面关于程序语言的叙述, 错误的是 ()。

- A. 脚本语言属于动态语言, 其程序结构可以在运行中改变
- B. 脚本语言一般通过脚本引擎解释执行, 不产生独立保存的目标程序
- C. `php`、`JavaScript` 属于静态语言, 其所有成分可在编译时确定
- D. `C` 语言属于静态语言, 其所有成分可在编译时确定

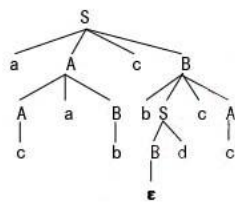
149. 已知某高级语言源程序 A 经编译后得到机器 C 上的目标程序 B , 则 ()。

- A. 对 B 进行反编译, 不能还原出源程序 A
- B. 对 B 进行反汇编, 不能得到与源程序 A 等价的汇编程序代码
- C. 对 B 进行反编译, 得到的是源程序 A 的变量声明和算法流程
- D. 对 A 和 B 进行交叉编译, 可以产生在机器 C 上运行的动态链接库

150. 函数调用时, 基本的参数传递方式有传值与传地址两种, ()。

- A. 在传值方式下, 形参将值传给实参
- B. 在传值方式下, 实参不能是数组元素
- C. 在传地址方式下, 形参和实参间可以实现数据的双向传递
- D. 在传地址方式下, 实参可以是任意的变量和表达式

151.由某上下文无关文法 $M[S]$ 推导出某句子的分析树如下图所示，则错误的叙述是（ ）。



- A. 该文法推导出的句子必须以“a”开头
- B. acabcbdcc 是该文法推导出的一个句子
- C. “ $S \rightarrow aAcB$ ”是该文法的一个产生式
- D. d 属于该文法的终结符号集

152.若一个程序语言可以提供链表的定义和运算，则其运行时的（ ）。

- A. 数据空间适合采用静态存储分配策略
- B. 数据空间必须采用堆存储分配策略
- C. 指令空间需要采用栈结构
- D. 指令代码必须放入堆区

153.以下关于编译系统对某高级语言进行翻译的叙述中，错误的是（ ）。

- A. 词法分析将把源程序看作一个线性字符序列进行分析
- B. 语法分析阶段可以发现程序中所有的语法错误
- C. 语义分析阶段可以发现程序中所有的语义错误
- D. 目标代码生成阶段的工作与目标机器的体系结构相关

154.将高级语言源程序翻译为机器语言程序的过程中常引入中间代码。以下关于中间代码的叙述中，错误的是（ ）。

- A. 不同的高级程序语言可以产生同一种中间代码
- B. 使用中间代码有利于进行与机器无关的优化处理
- C. 使用中间代码有利于提高编译程序的可移植性
- D. 中间代码与机器语言代码在指令结构上必须一致

155.以下关于 C/C++ 语言指针变量的叙述中，正确的是（ ）。

- A. 指针变量可以是全局变量也可以是局部变量
- B. 必须为指针变量与指针所指向的变量分配相同大小的存储空间
- C. 对指针变量进行算术运算是没有意义的
- D. 指针变量必须由动态产生的数据对象来赋值

156.许多程序设计语言规定，程序中的数据都必须具有类型，其作用不包括（ ）。

- A. 便于为数据合理分配存储单元
- B. 便于对参与表达式计算的数据对象进行检查
- C. 便于定义动态数据结构
- D. 便于规定数据对象的取值范围及能够进行的运算