一、单项选择题（共 20 题，每题 2.5 分，共 50 分）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 答案 | D | A | B | C | A | A | D | B | D | B |
| 题号 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 答案 | C | D | D | A | C | C | A | B | C | B |

1. 解析：在字符串中，乘法运算符用于生成重复的字符串。
2. 解析：在 Python 中，可以使用‘=’进行赋值，也可以使用‘-=’、‘+=’等进行赋值。
3. 解析：变量名可以包括字母、数字、下划线，但是数字不能做为开头。
4. 解析：保留字（称为关键字），指被编程语言内部定义并保留使用的标识符，程序员编写程序时不能定义与保留字相同的标识符。保留字不一定都是小写字母（例如布尔值：True 和 False），保留字不能作为变量名。
5. 解析：查看数据类型，可以使用内置函数 type()。如果变量 a = '3.0',变量 a 的数据类型是 str；

如果变量 a = 3.0,变量 a 的数据类型是 float； 如果变量 a = 3,变量 a 的数据类型是 int；

1. 解析：这里的 or 是逻辑运算符，其他的逻辑运算符还有 and 和 not。

or 可以对符号两侧的值进行或运算，两个值中只要有一个 True，就会返回 True。

这里的 10<=3 是不成立的，所以就需要继续验证, 5>2 是成立的，所以打印的值时 True。

1. 解析：首先对变量进行了赋值，a 为 2，b 为 3，c 为 7，变量 d 等于计算式 （b+c)-a\*a 的值， 也就是：（3+7)-2\*2 = 6。
2. 解析：在字符串中，加号用于拼接字符。
3. 解析：列表是序列类型，可以通过索引获得数据元素，这里 price[1:]表示从索引为 1 的位置截取后面全部的字符。
4. 解析：首先对变量进行了赋值，n1 为 3，n2 为 4，接着 if 条件语句判断的内容是：如果 n1<n2，就执行代码重新给 n1,n2 赋值。这里的 if 条件成立，所以 n1 重新赋值为 4，n2 重新赋值为 3。最后打印的 n1,n2 分别是 4 3。要得到选项 D 的结果，需要分别使用 print 打印变量 n1、n2，像是这样，print(n1),print(n2)。
5. 解析：双向分支有两个分支，当程序执行到 if...else...语句时，一定会执行 if 或 else 中的一个。这里的 if 条件判断语句的条件成立，所以就打印执行代码 print（‘产品合格’）。
6. 解析：input() 函数接受一个标准输入数据，返回为 string 类型，使用内置函数 int 把输入的内容转为整型，然后执行 if 条件判断语句。

选项 A，输入为 4，输出的结果应该是 4 元； 选项 B，输入为 5，输出的结果应该是 4 元； 选项 C，输入为 18，输出的结果应该是 8 元。

1. 解析：变量 a 是一个字符串，变量 b 通过 list 将序列 a 转换为一个列表。序列 a 中的每一个元素都是列表中的元素。
2. 解析：首先变量 n 等于 0，for 循环对列表[1，5，7，3]进行遍历，i 也就依次是列表的元素。在 for 循环中，如果变量 n 小于 i，就重新给 n 赋值为 i。

第一次循环，n=0，i=1，n<i 成立，n 重新赋值，n=1； 第二次循环，n=1，i=5，n<i 成立，n 重新赋值，n=5； 第三次循环，n=5，i=7，n<i 成立，n 重新赋值，n=7；

第四次循环，n=7，i=3，n<i 不成立，if 条件判断下的代码不执行。

1. 解析：这是一个 While True 无限循环，输入的值存储在变量 a 中，然后对 a 进行条件判断语句。如果 a 对 2 取模的值等于 0，就退出无限循环，程序停止运行，否则就打印 a\*2 的值。

第一次输入 5，5 对 2 取模等于 1，执行打印 5\*2=10； 第二次输入 3，3 对 2 取模等于 1，执行打印 3\*2=6；

第三次输入 4，4 对 2 取模等于 0，执行 break，程序停止运行；

1. 解析：异常捕捉可以使用 try/except 语句，题目中，if 条件判断语句对变量 s 进行判断，如果 s 大于 0， 就重新给 s 赋值，s 在之前的值上加 1，然后打印变量 s 的值。

try 语句按照如下方式工作；

首先，执行 try 子句（在关键字 try 和关键字 except 之间的语句）。如果没有异常发生，忽略 except 子句，try 子句执行后结束。

如果在执行 try 子句的过程中发生了异常，那么 try 子句余下的部分将被忽略。如果异常的类型和

except 之后的名称相符，那么对应的 except 子句将被执行。

如果一个异常没有与任何的 except 匹配，那么这个异常就会执行 else 子句，执行打印 print("出现未知错误")。

这里的变量 if 条件判断语句执行时，变量 s 未赋值，所以这是变量未赋值异常，except 子句中的异常的类型和当前的错误匹配，所以就会执行打印 print("变量未赋值")

1. 解析：pow()是 Python 的内置方法，方法返回 x 的 y 次方的值。pow(3.33,2):这个是表示 3.33 的 2 次幂。
2. 解析：sum()是 Python 的内置方法，sum() 方法对系列进行求和计算。这里只需要把列表的元素依次相加就可以了。
3. 解析：eval()是 Python 的内置方法，eval() 函数用来执行一个字符串表达式，并返回表达式的值。这里就执行 3.0 加上 2.0，结果为 5.0，需要注意的是这里的结果是浮点型哦。

选项 A 的执行代码是这样的：a = eval("3+2")

1. 解析：turtle(海龟)是 Python 重要的标准库之一，它能够进行基本的图形绘制。这里的 for 循环的次数是 4，每一次都执行让小海龟向前移动 100，然后左转 90 度。for 循环完成后，选用 hideturtle 方法把小海龟隐藏了起来。

二、多项选择题（共 5 题，每题 2 分，共 10 分）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题号 | 21 | 22 | 23 | 24 | 25 |
| 答案 | ACD | AB | AC | CD | BCD |

1. 解析：选项 B 的比较运算符是错误的，正确的小于等于运算符应该是 D 里的‘<=’。
2. 解析：选项 C，关于字符串转整型，正确的代码代码应该是 int('3')。

选项 D，字符串‘3.4’ 和字符串‘3.3’ 首先进行加法运行，运算后变成 ‘3.43.3’，关于字符串转浮点型，正确的代码形式应该是 float('3.4')。

1. 解析：要想获得输出结果包含 1，就需要程序 while 循环执行，选项 AC 都可以让程序重复执行，选项

B 和选项 D 程序不会执行。

1. 解析： len() 是 Python 的内置方法，len() 方法返回对象（字符、列表、元组等）长度或项目个数。选项 A，变量 a 是字符串，变量 b 的值是 5；

选项 B，变量 a 是浮点数，程序运行会报错； 选项 C，变量 a 是字符串，变量 b 的值是 3； 选项 D，变量 a 是列表，变量 b 的值是 3；

1. 解析：选项 A 错误，setheading(angle)设置当前朝向为 angle 角度。

三、判断题（共 10 题，每题 1 分，共 10 分）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 答案 | F | F | F | F | F | T | F | T | T | F |

1. 解析：在 Python 中可以用 Tab 和空格进行缩进，4 个空格相当于 1 个 Tab，两者不可以混用
2. 解析：多行注释可以使用三个双引号或者三个单引号将注释括起来。
3. 解析：input() 函数接受一个标准输入数据，返回为 string 类型。
4. 解析：在 Python 中，print()函数默认是换行的。
5. 解析：Python 代码可以在 IDLE 、交互式解释器中书写运行。

32. 解析：min() 方法返回给定参数的最小值，参数可以为序列。

1. 解析：语法错误指示软件的结构上有错误，导致不能被解释器解释或编译器编译。

逻辑错误可能是由于不完整或是不合法的输入所致，或者是无法生成、计算、或是输出结果需要的过程无 法执行。Python 解释器可以提示语法错误，不会提示逻辑错误。

四、编程题（共 2 题，每题 15 分，共 30 分）

1. 参考代码：

math=int(input("数学：")) chinese=int(input("语文：")) english=int(input("英语："))

score=math+chinese+english print(score)

if score>=270:

elif score>=240:

elif score>=180:

else:

测试样例：

print("优秀")

print("良好")

print("合格")

print("不合格")

|  |  |
| --- | --- |
| 输入：90  90  90 | 输出：270  优秀 |
| 输入：97  95  92 | 输出：284  优秀 |
| 输入：100  100  100 | 输出：300  优秀 |
| 输入：50  40  30 | 输出：120  不合格 |
| 输入：50  55  45 | 输出：150  不合格 |
| 输入：0  0  0 | 输出：0  不合格 |
| 输入：90  88  72 | 输出：260  良好 |
| 输入：60  60  60 | 输出：180  合格 |
| 输入：60  80  90 | 输出：240  良好 |

36. 参考代码

import random

lis = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

caipiao = random.sample(lis, 5)

caipiao = ''.join(caipiao) shuru = input("输入号码：") if caipiao == shuru:

print("奖金:10000 元") else:

i = 0

for s in shuru:

for c in caipiao: if s == c:

i += 1

print("彩票号:" + caipiao)

print("奖金:" + str(i \* 1000) + "元")

当生成的随机是 12345 时，使用以下测试用例测试样例：

输入：12345 输出：彩票号:12345

奖金:10000 元

输入：12468 输出：彩票号:12345

奖金:3000 元

输入：12341 输出：彩票号:12345

奖金:4000 元

输入：17901 输出：彩票号:12345

奖金:1000 元

输入：16578 输出：彩票号:12345

奖金:1000 元

输入：13452 输出：彩票号:12345

奖金:5000 元

输入：05978 输出：彩票号:12345

奖金:0 元

输入：86432 输出：彩票号:12345

奖金:3000 元

输入：25689 输出：彩票号:12345

奖金:1000 元

输入：87234 输出：彩票号:12345

奖金:3000 元