

一、单选题 (20%) 每题四个选项, 请在确切答案前用√标记

(1) 执行下列语句后的显示结果是什么?

```
>>> world=" world"
```

```
>>> print "hello" + world
```

- [A] helloworld [B] "hello" world
[C] hello world [D] 语法错

(2) 下列标识符中哪个是合法的?

- [A] i'm [B] _ [C] 3Q [D] for

(3) 执行下列语句后的显示结果是什么?

```
>>> from math import sqrt
```

```
>>> print sqrt(3)*sqrt(3) == 3
```

- [A] 3 [B] True [C] False [D] sqrt(3)*sqrt(3) == 3

(4) 设 s = " Happy New Year", 则 s[3:8]的值为:

- [A] 'ppy Ne' [B] 'py Ne' [C] 'ppy N' [D] 'py New'

(5) 算法是指:

- [A] 数学的计算公式 [B] 程序设计语言的语句序列
[C] 对问题的精确描述 [D] 解决问题的精确步骤

(6) type(1+2L*3.14)的结果是:

- [A] <type 'int' > [B] <type 'long' >
[C] <type 'float' > [D] <type 'str' >

(7) 以下何者是不合法的布尔表达式:

- [A] x in range(6) [B] 3=a [C] e>5 and 4==f [D] (x-6)>5

(8) 若 k 为整形, 下述 while 循环执行的次数为

```
k=1000
```

```
while k>1:
```

```
    print k
```

```
    k = k/2
```

- [A] 9 [B] 10 [C] 11 [D] 1000

(9) 选出对下列语句不符合语法要求的表达式:

```
for var in _____ :
```

```
    print var
```

- [A] range(0,10) [B] "Hello" [C] (1,2,3) [D] {1,2,3,4,5}

(10) 计算思维能力和下述其他方面能力相比, 何者更相近?

- [A] 阅读、写作和算术 [B] 逻辑推理
[C] 形象思维 [D] 技巧体操

二、填空题 (10%, 每小题 2 分)

- (1). Python 中调用 open 函数打开文件, 调用 close 函数关闭文件, 调用 _____ 函数可实现对文件内容的读取。
`read() or readline() or readlines()`
- (2). 表达式 $1/4+2.75$ 的值是 _____。
`2.75`
- (3). 高级程序设计语言必须由 _____ 或者 _____ 翻译成低级语言。
`编译器 解释器`
- (4). 给出 range(1, 10, 3) 的值: _____。
`[1, 4, 7]`
- (5). 请给出计算 $2^{31}-1$ 的 Python 表达式: _____。
`2**31-1`

三、是非题 (20%) 请在语句后的 [] 内填写标记 √ 或 X:

- (1) 函数 eval() 用于数值表达式求值, 例如 eval(2*3+1)。 [X]
- (2) 执行了 import math 之后即可执行语句 print sin(pi/2)。 [X]
- (3) Python 可以不加声明就使用变量。 [√]
- (4) Python 可以不对变量如 a 初始化就可在表达式如 b=a+1 中使用该变量。 [X]
- (5) 选择排序算法是一个时间复杂度为 $n\log n$ 算法。 [X]
- (6) Hanoi 塔的解法体现了分而治之(divide-and-conquer)方法的典型用途。 [X]
- (7) 一个函数中只允许有一条 return 语句。 [X]
- (8) 可用 open("file1", "r") 方式打开文件并进行修改。 [X]
- (9) Python 语言是面向对象的。 [√]
- (10) 存在一个程序判别一个 Python 程序的运行是否可以停止。 [X]

四、理解题 (24%) 写出下列程序的运行结果:

(1).

```
def func(s, i, j):
    if i < j:
        func(s, i + 1, j - 1)
        s[i],s[j] = s[j], s[i]

def main():
    a = [10, 6, 23, -90, 0, 3]
    func(a, 0, len(a)-1)
    for i in range(6):
        print a[i]
    print "\n"
```

main()

3
0
-90
23
6
10

(2).

```
i = 1
while i + 1:
    if i > 4:
        print "%d\n" % i
        i += 1
        break
    print "%d\n" % i
    i += 1
    i += 1
```

1
3
5

(3).

```
def foo(s):
    if s == "":
        return s
    else:
        return foo(s[1:]) + s[0]
```

print foo("Happy New Year")

.raeY weN yppaH

(4).

```
def func(a, n, m):
    if n == 0:
        return 1
    else:
        num = func(a, n/2, m)
        if n%2 == 0:
            return num * num % m
        else:
            return num * num * a % m
```

```
print func(5,6,7)
```

1

(5).

```
def foo(list,num):
    if num == 1:
        list.append(0)
    elif num == 2:
        foo(list,1)
        list.append(1)
    elif num > 2:
        foo(list,num-1)
        list.append(list[-1]+list[-2])
```

```
mylist = []
```

```
foo(mylist,10)
```

```
print mylist
```

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

(6).

```
def func(a,b):
    if (a < b):
        a, b = b, a
    r = a % b
    if r == 0:
        return b
    else:
        return func(b,r)
```

```
ans = func(342,84)
```

```
print ans
```

6

五、程序填空题 (10%)

(1). 打印 100~200 间的全部素数,输出格式为每行 10 个:

```
[----1----]
n = 0
for m in range(101, 201, 2):
    k = int(math.sqrt(m))
    for i in range([----2----], k+1):
        if m % i == 0:
            [----3----]
    if i == [----4----]:
        if n % 10 == 0:
            print "\n"
        print "%d " % m,
    [-- 5  --]
```

```
import math
2
break
k
n += 1
```

(2). 用选择排序算法对数组中的数据按由小到大排序。

```
def sort([----1----]):
    for i in range(n-1):
        [----2----]
        for j = range(i + 1, n):
            if [----3----]
                k = j
        a[k], a[i] = a[i], a[k]
def main():
    list = [12,5,61,8,11,32,111,94,37,55]
    [----4----]
    print "The sorted list:\n"
    for i in [----5----]:
        print list[i],
    print "\n"

main()
a,n
k = i
a[j] > a[k]
sort(list,len(list))
range(10)
```

六、程序设计题（16%，每小题8分）

(1) 编写函数，检测给定的二维数组中是否有重复数据（已知该数组中保存了正整数）：
check_data(mat)，如果没有重复，则函数返回真。要求算法的时间复杂度不大于 $O(n)$ 。

```
def expand_list(list,num):
    if len(list) > num:
        return
    for i in range(len(list),num):
        list.append(0)

def check_data(list):
    count = []
    expand_list(count,10)
    for line in list:
        for var in line:
            if var > len(count):
                expand_list(count,var+1)
            if count[var] == 0:
                count[var] += 1
            else:
                return False
    return True
```

(2) 设有一个背包能承受重量 $s(s>0)$ ，又有 $n(n\geq 1)$ 件物品，其重量列表为 $w=[w_1, w_2, \dots, w_n]$ 。要从物品中挑选若干件放入背包，使得放入的物品重量之和正好是 s 。试设计递归函数 $f(w, s, n)$ ：当有解时返回 True，并显示所选物品是哪些；无解时返回 False。[提示：递归之处在于 $f(w, s, n)$ 的真假等于 $f(w, s-w_n, n-1)$ or $f(w, s, n-1)$]

```
def f(w,s,n):
    if s==0:
        return True
    elif (s<0) or (s>0 and n<1):
        return False
    elif f(w,s-w[n-1],n-1):
        print w[n-1],
        return True
    else:
        return f(w,s,n-1)
```