

2025 新教材技术高考第一轮

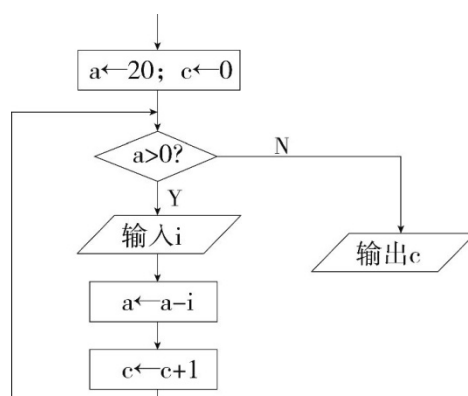
专题二 算法与程序基础

考点过关练

考点一 算法的概念及描述

考点二 算法的控制结构

1.(2024 届杭州质量检测,7)某算法的部分流程如图所示,执行这部分流程,若输入 i 的值为序列[2,3,4,5]中的随机数,则下列说法正确的是()



A.变量 c 的最大值为 11

B.“ $a > 0$?”至少执行 5 次

C.程序结束时变量 a 一定等于 0

D.只能使用 while 语句实现该循环结构

2.关于算法的重要特征,下列说法错误的是()

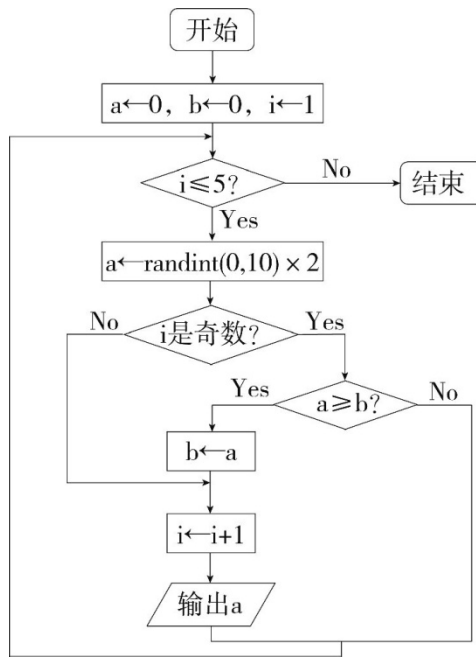
A.有穷性:算法必须能在执行有限个步骤之后终止

B.确定性:算法中的每一次运算都既有明确的定义,也可具有二义性

C.输入项:一个算法有 0 个或多个输入,以刻画运算对象的初始情况,所谓 0 个输入是指算法本身给出了初始条件;输出项:算法一定要有输出

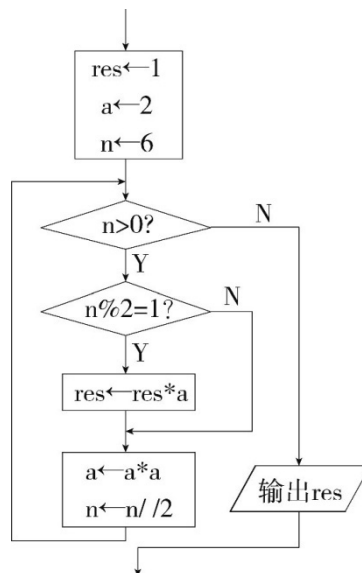
D.可行性:算法中执行的任何计算都可以在有限时间内完成(也称为有效性)

3.(2022 精诚联盟联考,7)某算法流程图如图所示,已知 `randint(0, 10)` 函数可以产生闭区间[0, 10]内的随机整数,执行完该流程图后,输出结果可能是()



- A.6 6 6 6 6
- B.10 12 16 18 22
- C.0 2 4 5 8
- D.12 8 10 16 20

4.(2023 海宁适应性考试,7)某算法的部分流程图如图所示,若用 Python 语言描述,则下列选项正确的是()



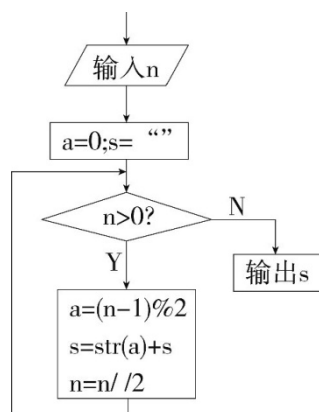
- A.res=1;a=2
- for n in range(6,0,-2):
- if n%2==1:
- res*=a

```

    else:
        a*=a
        n//=2
B.res=1;a=2;n=6
while n>0:
    if n%2==1:
        res*=a
    else:
        a*=a
        n//=2
C.res=1;a=2
for n in range(6,0,-2):
    if n%2=1:
        res*=a
        a*=a
        n//=2
print(res)
D.res=1;a=2;n=6
while n>0:
    if n%2==1:
        res*=a
        a*=a
        n//=2
print(res)

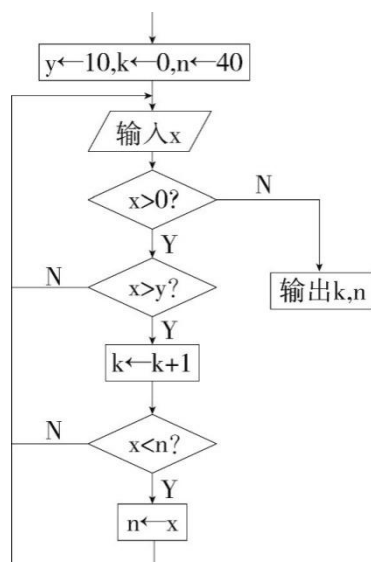
```

5.(2023 东阳模拟,7)某算法的部分流程图如图所示。执行这部分流程,若输入 n 的值为 55,则输出 s 的值为()



- A.110111 B.000100
C.001000 D.111011

6.(2023 金丽衢联考,7)某算法的部分流程图如图所示,执行这部分流程图,若输入的值依次为 7, 20,8, 15, 10, 40, -1,则输出结果为()



- A.4,40 B.3,10
C.1,20 D.3,15

考点三 用算法解决问题的一般过程

考点四 用计算机编程解决问题的一般过程

1.使用算法解决实际问题时,首先应该做的是()

- A.设计算法 B.抽象与建模
C.编写程序 D.描述算法

2.任何一个算法都要使用的控制结构是()

- A.循环结构 B.树型结构

C.分支结构 D.顺序结构

3.要编程判断一个正整数是不是素数(质数),最适合使用的控制结构是()

A.顺序结构 B.分支结构

C.选择结构 D.循环结构

4.小王想设计一个文件加密的程序,为此她制订了如下的工作过程,你认为最恰当的是()

A.设计算法、编写程序代码、调试程序、得到答案

B.分析问题、编写程序代码、运行程序、得到答案

C.分析问题、设计算法、编写程序代码、调试运行

D.分析问题、设计程序、编写程序代码、运行程序

5.解决问题的一般过程中,需要先分析问题,对问题进行抽象与建模。以下不属于分析问题阶段的是()

A.查找与问题相关的资料

B.寻找已知与未知的联系

C.明确问题所涉及的相关知识

D.设计问题求解方案

6.(2023 舟山期末,7)下列关于算法的说法正确的是()

A.算法解决问题的一般过程依次为“设计算法、抽象与建模、描述算法”

B.数据、运算和控制转移是算法三大要素

C.任何算法都必须要有至少一个输入数据和一个输出数据

D.同一种算法只能用一种方法表示

考点五 Python 语言程序设计

考点六 简单算法及其程序实现

1.(2023 余姚期中,9)若 x 是整型变量,下列选项中与表达式 $\text{not}(x \geq 1 \text{ and } x < 5)$ 等价的是()

A. $x < 1 \text{ and } x > 5$ B. $\text{not } x \geq 1 \text{ or } x < 5$

C. $x < 1 \text{ and } x < 5$ D. $x < 1 \text{ or } x \geq 5$

2.(2023 杭州 S9 联盟期中,5)已知字典 $\text{dic}=\{\text{False}: 0, \text{True}: 1\}$,执行语句 $\text{print}(\text{dic}["\text{hi}" \text{ in } "\text{height}" \text{ or } "3" > "21"])$ 后的结果是()

A.False B.0

C.True D.1

3.(2023 杭州“六县九校”期中,11)某 Python 程序如下:

```
d={"A":1,"B":2,"C":3,"D":4,"E":5}
```

```
a=d["E"]
```

```
b=d["D"]
```

```
c=d["B"]
```

```
s=a+b+c
```

```
print(s)
```

程序运行后,输出的结果是()

A.8 B.9 C.10 D.11

4.(2023 宁波期末,10)某停车场的收费标准为30 分钟内(含)免费,3 小时内(含)5 元,超过 3 小时每增加 1 小时加收 1 元,停车不足 1 小时的按 1 小时计算。”已知停车时长为 t 分钟,则下列 Python 程序段中不能正确计算停车费 m 的是()

A.if t<=30:

```
    m=0
```

elif t<=3*60:

```
    m=5
```

else:

```
    m=5+(t-3*60+59)//60
```

B.if t>30:

```
    m=5
```

```
    if t>3*60:
```

```
        m=5+(t-3*60+59)//60
```

else:

```
    m=0
```

C.m=5

if t>3*60:

```
    m=5+(t-3*60+59)//60
```

elif t<=30:

```
    m=0
```

D.m=0

if t>3*60:

m=5+(t-3*60+59)//60

if t>30:

m=5

5.(2023 杭州“六县九校”期中,12)已知 $d1=\{\text{"地点":["杭州","金华","温州"],\text{"牌号":["a","b","c"],\text{"编号":[1,2,3]}\}}$, $s=\text{"VALUE"}$,下列表达式值与其他三项不相同的是 ()

A.ord(s[1])+3//10*10

B.len(d1["地点"][0])**6+1

C.74-d1["编号"][2]**2

D.int((ord(d1["牌号"][1])-32)/10)*10

6.(2023 余姚期中,12)有下面一段程序:

a=[9,5,2,3,0,-9,5,0,7,-10]

s,i=0,0

while a[i]!=0 and i < len(a):

s+=1

i+=1

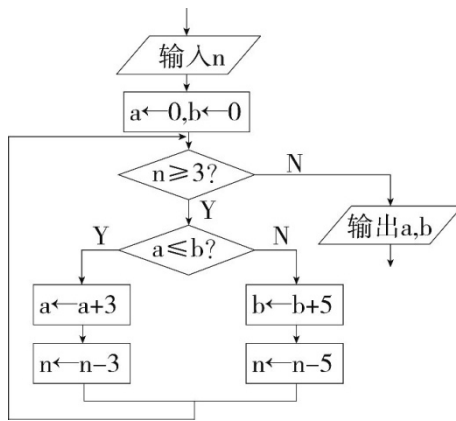
运行该程序后 s 的值为()

A.4 B.5 C.8 D.10

专题综合练

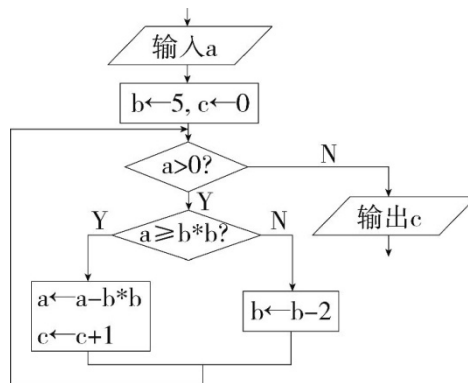
题组一

1.(2021 浙江 6 月选考,8,2 分)某算法的部分流程图如图所示,执行这部分流程,若输入 n 的值为 20,则输出 a,b 的值分别是()



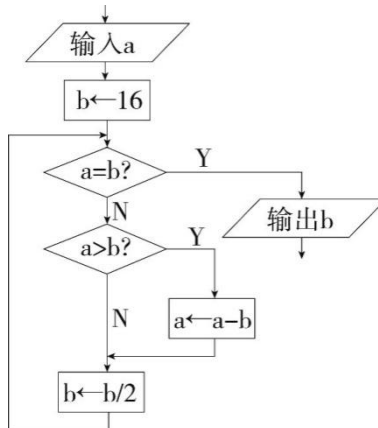
- A.12,10 B.10,9
C.9,15 D.9,10

2.(2020 浙江 7 月选考,8,2 分)某算法的部分流程图如图所示。执行这部分流程,若输入 a 的值为 36,则输出 c 的值为()



- A.2 B.3 C.4 D.5

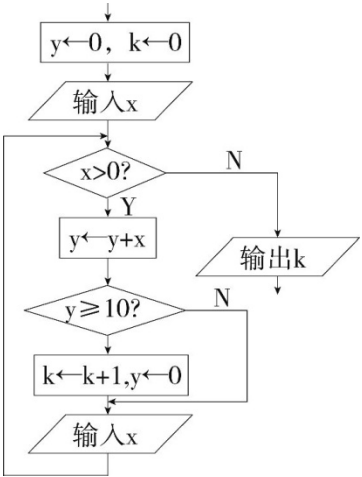
3.(2020 浙江 1 月选考,8,2 分)某算法的部分流程图如图所示。执行这部分流程,若输入 a 的值为 22,则输出 b 的值为()



- A.0 B.1 C.2 D.16

4.(2023 浙江 1 月选考,7,2 分)

某算法的部分流程图如图所示,执行这部分流程,若输入 x 的值依次为 10,7,8,12,0, 则输出 k 的值是()



A.2 B.3 C.4 D.5

5.(2024 届杭州 S9 联盟期中,6)下列语句的功能与其他三项不一样的是()

A.if a>b:

 flag=True

flag=False

B.flag=False

if a>b:

 flag=True

C.if a<=b:

 flag=False

else:

 flag=True

D.flag=a>b

6.(2024 届嘉兴基测,8)下列 Python 表达式运算结果为 True 的是()

A.3**2<8 and 2%5>1

B.25//4>4 and 3/2<1

C.not 7<8 or 9>6+2

D.not(2**3<9 or 6%4<2)

7.(2024 届七彩联盟联考,12)有如下 Python 程序:

```
import random
```

```
s=input()
```

```
k=random.randint(1,len(s)-1)
```

i=0

```
while k>0 and i<len(s)-1:
```

```
    if s[i]>s[i+1]:
```

```
        k-=1
```

```
        s=s[:i]+s[i+1:]
```

```
        if i>0:
```

```
            i-=1
```

```
    else:
```

```
        i+=1
```

```
if k>0:
```

```
    s=s[: len(s)-k]
```

若输入的 s 值为"8561324",则执行该程序,输出 s 的值不可能为()

A.51324 B.124 C.132 D.1324

8.(2024 届嘉兴基测,9)有如下 Python 程序段:

```
s="A-b2c";ch=" "
```

```
list=[4,3,1,5,2]
```

```
for i in range(len(s)):
```

```
    if "A"<=s[i]<="Z":
```

```
        ch=ch+chr(ord(s[i])+list[i])
```

```
    elif "a"<=s[i]<="z":
```

```
        ch=ch+chr(ord(s[i])-list[i])
```

```
    else:
```

```
        ch=s[i]+ch
```

```
print(ch)
```

该程序运行后,输出的结果是()

A.2-Eaa B.2-ECE

C.E-c2e D.E-a2b

9.(2024 届名校协作体联考,12)有如下 Python 程序段:

```
import random
```

```
a=[]
```

```
for i in range(6):
```

```
a.append(random.randint(1,10))
```

```
for i in range(5):
```

```
if i%2==0 and a[i]<a[i+1]:
```

```
    a[i],a[i+1]=a[i+1],a[i]
```

```
else:
```

```
    a[i]-=1
```

```
print(a)
```

运行该程序后,输出结果不可能是()

A.[7,3,3,4,9,4]

B.[8,8,10,5,7,8]

C.[9,4,9,4,8,7]

D.[7,2,9,7,4,5]

10.(2024 届强基联盟统测,12)列表 a 有 $n(n \geq 2)$ 个正整数,相邻两数的差值为 1,0 或 -1。求相邻两数具有相同差值的连续子序列中,子序列元素和的最大值。实现该功能的程序段如下,加框处应填写的正确代码是()

```
state=a[1]-a[0]
```

```
maxv=t1=t2=a[1]+a[0]
```

```
for i in range(2,n):
```

```
    f=a[i]-a[i-1]
```

```
print(maxv)
```

A.if state==f:

```
    t1+=a[i]
```

```
t2=t1
```

```
else:
```

```
    state=f
```

```
t2=t1
```

```
t1=a[i-1]+a[i]
```

```
t2=max(t1,t2)
```

```
if t2>maxv:
```

```
maxv=t2
B.if state==f:
    t1+=a[i]
    t2=t1
elif state !=f:
    state=f
    t2=t1
    t1=a[i-1]+a[i]
    t2=max(t1,t2)
elif t2>maxv:
    maxv=t2
C.if state==f:
    t1+=a[i]
    t2=t1
if state !=f:
    state=f
    t2=t1
    t1=a[i-1]+a[i]
    t2=max(t1,t2)
elif t2>maxv:
    maxv=t2
D.if state==f:
    t1+=a[i]
    t2=t1
else:
    state=f
    t2=t1
    t1=a[i-1]+a[i]
    t2=max(t1,t2)
if t2>maxv:
```

```
maxv=t2
```

11.(2024 届强基联盟统测,9)有如下程序段:

```
a=[30,10,20,11,12,50]
d=[0]*6 ; max=0 ; n=len(a)
for i in range(1,n):
    for j in range(i):
        if a[j]<a[i]:
            d[i]+=1
    if d[i]>max:
        max=d[i]
```

执行程序后,变量 max 的值是()

A.2 B.3 C.4 D.5

12.(2024 届杭州质量检测,10)列表 d 有 n 个整型元素,输出列表中最长连续递增序列的长度。小明编写的 Python 程序段如下:

```
k=1;lmax=1;n=len(d)
for i in range(1,n):
    if d[i]>d[i-1]:
        k+=1
    else:
        if k>lmax:
            lmax=k
        k=1
```

```
print(lmax)
```

该程序段在某些情况下无法得到符合题意的结果,下列 4 组数据中能测试出这一问题的是()

A.[1,4,5,1,2] B.[1,2,3,4,1]
C.[1,5,1,2,3] D.[5,4,3,2,1]

13.(2023 宁波期中,8)已知字典 d={'名称':'orange','数量':7,'售价':6.7,'利润':6.78},则下列 Python 表达式中值最大的是()

- A.len(d['名称']) B.d['数量']
C.round(d['利润'],1) D.int(d['售价'])

14.(2023 衢州质量检测,11)有如下 Python 程序段:

```
def sm(a):  
    if len(a)==1:  
        return a[0]  
    else:  
        return a[0]+sm(a[1:])
```

```
a=[7,8,9,1,5,6]
```

```
print(sm(a))
```

程序运行后,函数 sm 被调用的次数是()

- A.5 B.6 C.7 D.12

15.(2022 浙江 7 月学考,10,3 分)某 Python 程序如下:

```
p="Tel-114"
```

```
c=" "
```

```
for ch in p:
```

```
    if ch>="0" and ch<="9":
```

```
        c+=str(9-int(ch))
```

```
    elif ch>="a" and ch<="z":
```

```
        c+=chr(ord(ch)- ord("a")+ord("A"))
```

```
    else:
```

```
        c+=ch
```

```
print(c)
```

程序运行后,输出的结果是()

- A.tEL-885 B.tEL-114
C.TEL-114 D.TEL-885

16.(2023 三校联考,10)有如下 Python 程序段:

```
a=[]
```

```
b=[1,5,4,1,5]
```

```
c={1:3,2:4,3:5,4:1,5:2}
```

```
for i in b:
    if i not in a:
        a.append(c[i])
print(a)
```

运行该程序段后,列表 a 的值是()

- A.[1,5,4] B.[3,2,1]
C.[3,2,1,2] D.[3,2,1,3]

17.(2023 嘉兴基测,12)有如下 Python 程序段:

```
import random
nums=[0]*6
for i in range(6):
    nums[i]=random.randint(1,10)
i=0
j=len(nums)-1
while i<j:
    if nums[i] %2==0 and nums[j]%2==1:
        nums[i],nums[j]=nums[j],nums [i]
    elif nums[i] %2==1:
        i=i+1
    elif nums[j] %2==0:
        j=j-1
print(nums)
```

执行该程序段后,下列输出结果中不可能的是()

- A.[1,3,1,5,1,7]
B.[1,4,3,6,5,8]
C.[1,3,5,7,9,10]
D.[6,4,2,8,2,10]

18.(2023 余姚期中,17)

学考成绩与学考等级折算方法如下:“A”等折算成 10 分,“B”等折算成 9 分,“C”等折算成 7 分,“D”等折算成 4 分。现编写一个能实现将学考等级折算成相应分数的 Python 程序,程序运行结果如图所示:

请输入各门功课的学考等级:ABABACBAAA

你的学考成绩折算为:94

请回答下列问题:

(1)若输入"aaabbaac",该程序_____ (填:能/不能) 计算出相应的折算成绩。

(2)请在程序划线处填上合适的代码。

```
level=input("请输入各门功课的学考等级:")
```

```
dic={"A":10,"B":9,"C":7,"D":4}
```

```
score=0
```

```
for i in range(len(level)):
```

```
    score=score+_____①_____
```

```
print("你的学考成绩折算为:"+_____②_____)
```

19.(2022 浙江 7 月学考,12,10 分)查找与替换。从键盘上分别输入要查找和替换的字符串,对文本文件进行查找与替换,替换后保存到新的文本文件中。

完成查找与替换功能的思路是:首先可从待检索文本文件“in.txt”逐行读取文本内容到列表 text,然后从键盘上输入查找的字符串 key 和替换的字符串 new,对列表 text 中的元素逐个进行查找并替换,结果保存到列表 result,最后将 result 写入文件“out. txt”。

(1)主程序。

```
text=readfile("in.txt")      #读入文件
```

```
key=input("请输入要查找的字符串:")
```

```
new=input("请输入要替换的字符串:")
```

```
result=[]
```

```
for line in text:
```

```
    newline=replace(key, new, line)#替换
```

```
    result.append(newline)#添加到列表
```

`writefile("out.txt",result)#写入文件`

该程序段采用的算法是_____ (单选,填字母:A.解析算法 / B.枚举算法)。

(2)读写文本文件。如下的 readfile 函数,逐行读取文本文件数据存入列表并返回。
请在划线处填入合适的代码。

```
def readfile(filename):  
    f=open(filename,encoding="utf-8")#打开文件  
    text=[]  
    line=f.readline()#从文件中读取一行  
    while line:  
        text.append(line)#添加到列表  
        line=f.readline()  
    f.close()  
    return _____  
def writefile(filename,text):  
    #将 text 写入 filename 文件,代码略
```

(3)查找字符串。如下的 findstr 函数,在字符串 line 中从 begin 位置开始查找 key
在字符串 line 中的位置,请在划线处填入合适的代码。

```
def findstr(key,line,begin):  
    for i in range(begin, len(line) - len(key)+ 1):  
        if _____:  
            return i  
    return -1
```

(4)替换字符串。如下的 replace 函数,在字符串 line 中检索所有的字符串 key 并替
换为 new,请在划线处填入合适的代码。

```
def replace(key,new,line):  
    begin=0  
    while begin < len(line) - len(key)+ 1:  
        pos=findstr(key,line,begin)  
        if pos== -1:  
            _____  
        else:  
            line=line[0:pos]+ new+ line[pos+ len(key) :len(line)]
```

```
begin=pos+ len(key)
```

```
return line
```

20.(2023 浙江 6 月选考,13,7 分)某仓库有一排连续相邻的货位,编号依次为 0~n-1,用于放置 A、B 两种类型的箱子,A 型箱子占 2 个相邻货位,B 型箱子占 1 个货位。编写程序,根据已完成的放置或搬离操作,输出空货位数及还可以放置 A 型箱子的最多数量(不移动已放置的箱子)。请回答下列问题:

(1)若 n 为 10,开始时货位全空,经过如表所示的放置或搬离操作后,不移动已放置箱子的情况下,还可放置 A 型箱子的最多数量为_____个。

箱子类型	操作类型	货位编号
B	放置	5
A	放置	2,3
B	放置	0
A	放置	7,8
A	搬离	2,3

(2)实现上述功能的部分 Python 程序如下,请在划线处填入合适的代码。

#读取货位总数,存入 n,代码略。

```
cnt1=n
```

```
lst=[0]*n
```

#货位状态,0 表示对应的货位为空

```
while True:
```

#读取本次已操作的数据:箱子类型、操作类型、货位编号起始值,存入 t、d 和 s,代码略

```
if t=='A':
```

```
w=2
```

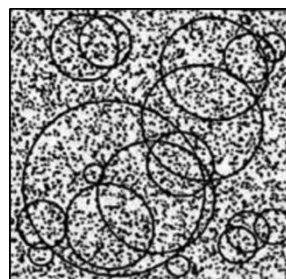
```
_____①_____:
```

```

        w=1
    else:    #t 不是'A'或'B'时退出循环
        break
    if d=='P':    #d 为 P 时表示放置,否则表示搬离
        _____②_____
    else:
        cnt1+=w
    lst[s]=1-lst[s]
    if t=='A':
        lst[s+1]=1-lst[s+1]
    i, cnt2=0,0
    while i< n-1:
        if lst[i]==0 and lst[i+1]==0:
            _____③_____
            cnt2+=1
        i+=1
    print('当前空货位数:',cnt1,'还可放置 A 型箱子的最多数量:',cnt2)

```

21.(2022“9+1”联盟期中,14)蒙特卡罗方法是一类基于概率,通过使用随机数来解决某些计算问题的方法。如图所示,在平面上有 k 个圆,可以使用蒙特卡罗方法求这些圆所占的近似总面积。(注:重叠部分仅计算一次,不重复计算)



具体算法如下:

- (1)建立一个直角坐标系,测得每个圆的坐标位置和半径;
- (2)用一个矩形包围所有的 k 个圆,设该矩形的长、宽分别为 m 和 n ;

(3)在矩形范围内产生 v 个随机点;

(4)统计落在圆内或圆弧上的随机点的数量,并存入变量 cnt ;

(5)根据 cnt 占有所有随机点的比例,计算所有圆的近似总面积 $S=矩形面积\times(cnt\div v)$ 。

请在划线处填入合适的代码。

提示:平面上两个坐标点 (x_1,y_1) 和 (x_2,y_2) 的距离 $=\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$ 。

```
from math import sqrt
```

```
import random
```

```
#测量矩形的长度、宽度、圆的个数,存入变量 m、n、k,代码略
```

```
v=int(input()) #读入随机点的数量,数量越多,精度越高
```

```
cir=[]
```

```
for i in range(k):
```

```
#依次测量每个圆的圆心(x,y)及半径 r,存入 cir 列表
```

```
    cir.append([x,y,r])
```

```
def dis(x1,y1,x2,y2): #计算两个坐标点的距离,参考题目中给出的公式
```

```
    d=_____①_____
```

```
    return d
```

```
def check(x,y):
```

```
    flag=False
```

```
    for i in range(k):
```

```
        if dis(x,y,cir[i][0],cir[i][1])<=cir[i][2]:#判断点是否在圆内或圆上
```

```
        flag=True
```

```
break
```

```
_____②_____
```

```
cnt=0
```

```
for i in range(v):
```

```
    x=random.uniform(0,n) #产生当前随机点的 x 坐标
```

```
y=random.uniform(0,m) #产生当前随机点的 y 坐标
```

```
if check(x,y):
```

```
    cnt+=1
```

```
_____③_____
```

```
print(round(s,2)) #输出近似总面积,四舍五入保留两位小数
```

22.(2023 学军中学月考,16)扫描算法是一种电梯的调度算法(非实际操作),电梯在最底层(1 楼)和最顶层(20 楼)之间连续往返扫描并运行,在运行过程中优先处理与当前电梯运行方向相同的请求,比如:当前电梯在 3 楼,方向为向上,此时有 3 个人请求使用电梯:7 楼去 16 楼、2 楼去 9 楼、6 楼去 1 楼,则电梯先向上运行,依次在 7 楼和 16 楼停靠;然后再向下运行,依次在 6 楼和 1 楼停靠;最后再向上运行,依次在 2 楼和 9 楼停靠。小明编写程序实现这个调度算法,运行界面如图,请回答下列问题。

当前楼层: 3 方向: 向上
向上, 停靠楼层: 7 16
向下, 停靠楼层: 6 1
向上, 停靠楼层: 2 9

(1)若电梯当前在 3 楼,方向为向上,收到的请求是:7 楼去 16 楼、6 楼去 1 楼、2 楼去 9 楼、14 楼去 1 楼、10 楼去 15 楼,则第一趟向上运行依次停靠的楼层是(按停靠顺序填写数字)。

(2)实现算法的 Python 程序如下,请在划线处填入合适的代码。

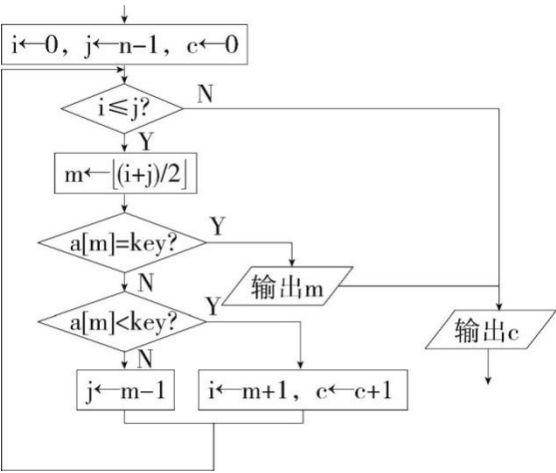
```
def scan(now,d):    #scan 函数的功能:从当前楼层开始,按当前运行方向扫描一趟
    for i in range(len(ask)):
        tmp=(d==1 and ask[i][0]>=now) or (____①____)
        if ask[i][2]==d and not flag[i] and tmp:
            stop[ask[i][0]]=1
    stop[ask[i][1]]=1
    flag[i]=True
ask=[[7,16,0],[6,1,0],[2,9,0],[14,1,0],[10,15,0]]
flag=[False for i in range(len(ask))]
direct={1:"向上",-1:"向下"}
now=3
d=1
print("当前楼层:",now,"方向:",direct[d])
```

```

for i in range(len(ask)):
    ②
if t>0:
    ask[i][2]=1    # t 表示该请求的电梯运行方向
elif t<0:
    ask[i][2]=-1    #t 表示该请求的电梯运行方向
else:
    flag[i]=True
while False in flag:
    stop=[0]* 21    #标记 1~20 各楼层是否停靠
scan(now,d)
print("n", direct[d], ",停靠楼层: ", end=" ")
if d==1:
    st=1
    ed=21
    now_next=20
else:
    st=20
    ed=0
    now_next=1
for i in range(st, ed, d):
    if stop[i]==1:
        print(i, end=" ")
now=now_next
    ③

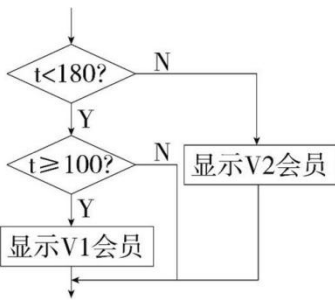
```

1.(2024 浙江 1 月选考,10,2 分)某算法的部分流程图如图所示,若 n 的值为 7, key 的值为 78,数组元素 $a[0]$ 至 $a[n-1]$ 依次存放 7,12,24,36,55,78,83,执行这部分流程后,输出 c 的值为 ()

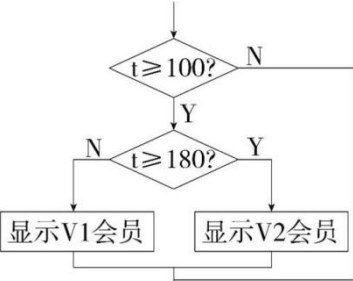


- A.0 B.1 C.2 D.3

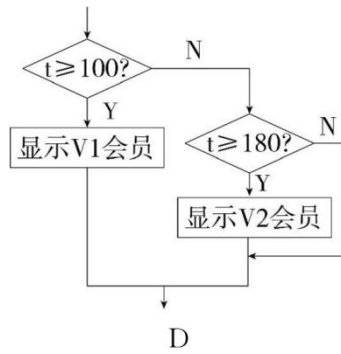
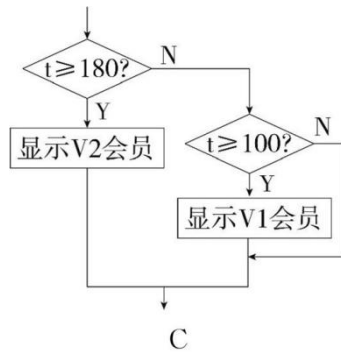
2.(2023 浙江 7 月学考,6,3 分)用算法流程图描述“消费积分累计 100 分,升级为 V1 会员;消费积分累计 180 分,升级为 V2 会员”的问题,用 t 表示顾客的消费积分,下列选项不正确的是()



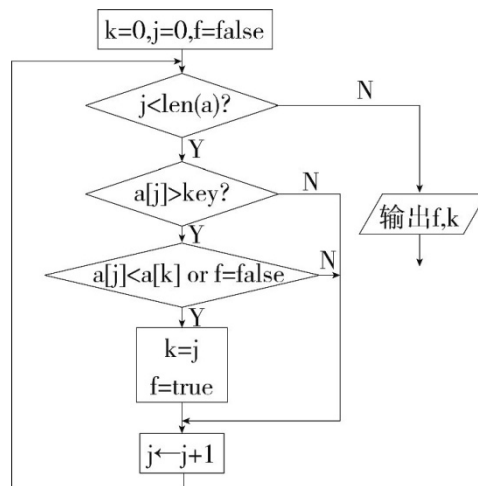
A



B

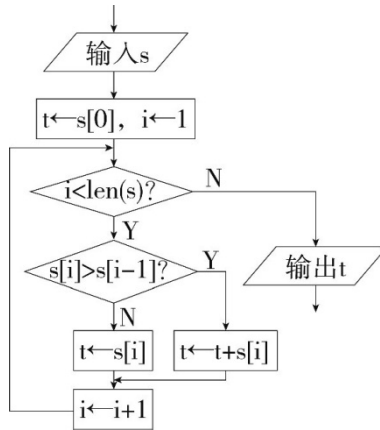


3.(2023 桐乡月考,16) 某算法的部分流程图如图所示,若列表 a 的值为 $[162,184,171,166,177,179]$, key 的值为 170,执行这部分流程后, f 、 k 的值分别为()



- A.false,0 B.true,1
C.true,2 D.true,3

4.(2023 十校联盟联考,6)某算法的部分流程图如图所示。执行这部分流程,若输入 s 的值为“0152234512”,则输出结果为()



A.3 B.12 C.015 D.22345

5.(2024 届三校联考,8)已知变量 $s="2021china"$,则表达式 $s[-1]+s[2]*2$ 的值是()

A."a4" B."a22" C."n22" D."a0"

6.(2023 温州适应性考试,7)已知列表 $a=[2,3,4,5]$,要将列表 a 更新为 $[1,2,3,4,5]$,则下列可实现该功能的操作是()

A. $a="1"+a$ B. $a.append(1)$

C. $a=[1]+a$ D. $a=1+a$

7.(2022 浙江 7 月学考,6,3 分)用 Python 算法控制结构描述“您所在街道 25 分钟后小雨,50 分钟后雨停”,设 t 为距离天气预报发布时间的间隔(单位:分钟),下列选项正确的是()

A.if $t>50$:

 print("雨停")

elif $t>25$:

 print("小雨")

B.if $t>50$:

 print("雨停")

else:

 print("小雨")

C.if $t>25$ and $t\leq 50$:

 print("小雨")

else:

 print("雨停")

D. if t>25:

```
    print("小雨")
```

elif t>50:

```
    print("雨停")
```

8.(2023 杭州地区重点中学期中,9)左右对称的字符串称回文串。若要判断字符串 s 是不是回文串,可以先找到 s 的中间位置 $mid=len(s)//2$,再采用切片操作取出 s 的左半部分 $left=s[0:mid]$,然后取出 s 的右半部分并反转,存储在变量 right 中,若 left 与 right 相等,则为回文串。则下列表达式能正确表达变量 right 值的是()

A. $s[mid:][::-1]$

B. $s[mid+1:][::-1]$

C. $s[len(s)-1:mid:-1]$

D. $s[-1:-mid-1:-1]$

9.(2023 浙江 7 月学考,10,3 分)某 Python 程序如下:

```
p=")!@#$$%^&*("
```

```
c="cra2edu"
```

```
t=" "
```

```
for i in range(len(c)//2+1):
```

```
    if i%2==0:
```

```
        t+=c[len(c)-i-1]
```

```
    elif c[i]>="0" and c[i]<="9":
```

```
        t+=p[int(c[i])]
```

```
    else:
```

```
        t+=c[i]
```

```
print(t)
```

程序运行后,输出的结果是()

A. ude@

B. cda#

C. ure@

D. arc#

10.(2024 浙江 1 月选考,11,2 分)若字符串 s 的值为"abcde",执行如下程序段后,变量 res 的值不可能是()

```
from random import randint
```

```
res=" "
```

```

i,j=0,len(s)-1
while i<len(s) and j>=i:
    if randint(0,1)==0:      #randint
        (0,1)随机生成 0 或 1
        res+=s[i]
        i+=1
    else:
        res+=s[j]
        j-=1

```

- A."abced" B."aecbd"
 C."aedbc" D."edcba"

11.(2023 十校联盟联考,11)有下列 Python 程序段:

```

dict={"a":["apple",["abandon","ant"]],"b":["banana","bee","become"],"c":["cat","com
e"],"d":"down"}
s="abcd"
cnt=0
for i in s:
    cnt+=len(dict[i])
print(cnt)

```

该程序段运行后,输出结果是()

- A.8 B.9
 C.11 D.12

12.(2024 届新阵地联盟联考,10)某 Python 程序如下:

```

n=0
for i in range(32):
    m=i;x=0
    while m>0:
        x+=m%2
        m//=2
    if x==3:

```

```
n+=1
```

```
print(n)
```

程序运行后,输出的结果是()

A.3 B.5 C.10 D.32

13.(2023 杭州“六县九校”期中,13)“最强大脑”游戏问题的算法分析:显示游戏规则,显示物品以及编号,屏幕停留 10 秒;随机出题,给出物品,回答编号;判断回答是否正确,如果正确,则答对的题数加 1。重复以上步骤,直到答完 5 道题;显示答对的题数。划线处应填入的代码为()

```
import random;import time;import os
```

```
print("你好,现在你有 10 秒钟的时间记忆下列物品及其编号")
```

```
things=["苹果","香蕉","橙子","梨子","猕猴桃","柚子","猴魁","铁观音","彩蛋","复活节"]
```

```
for i in range(10):
```

```
    print(i,":",_____①) #在屏幕上显示编号及物品名称
```

```
    time.sleep(10)#延时 10 秒
```

```
    os.system("cls")#清屏幕,隐去编号及物品
```

```
    n=0#记录答对的题数,初值为 0
```

```
    t2=random.sample(_____②,5)#随机抽出 5 个物品
```

```
    for i in t2:#出 5 题
```

```
        ans=int(input(i+ "的编号是:")) #输入编号答题
```

```
        if i==_____③:
```

```
            n=n+1 #如果回答正确,答对的题数加 1
```

```
print("\n 你一共答对了",n,"次") #屏幕显示答对的题数
```

```
input("\n 按回车键结束程序")
```

A.①things[i] ②things

③things[ans]

B.①things ②things

③things[i]

C.①things[i] ②things[i]

③things[ans]

D. ①things[ans] ②things[ans]

③things[i]

14.(2024 届七彩联盟联考,13)某公路由于长期没有维修,路上出现了很多个坑。为了尽快填补好这些坑,交通管理部门决定对 m 处地段采取交通管制。将该公路看成一条直线,坑就是直线上的坐标点,坑所在的路段需要封控管制。例如某管制路段 2~4,需封控 2、3、4 路段。交管部门为了减少管制路段的长度,希望将这 n 个坑分成 m 段(一段可以只有一个坑),使得这 m 段公路的总长度最小。请你根据 n 个坑的位置(位置已按照从小到大进行排序),计算管制路段最小的总长度。代码运行效果如图所示。

```
路段数量:4
坑的坐标依次为:3, 4, 6, 8, 14, 15, 16, 17, 21, 25, 26, 27, 30, 31, 40, 41, 42, 43
维修管制的路段依次为:
3 ~ 8
14 ~ 17
21 ~ 31
40 ~ 43
管制总长度为 25
```

请回答下列问题:

(1)上图所示的例子中,若将路段数量修改为 5,则管制路段总长度为_____。

(2)实现上述功能的 Python 程序如下,请在划线处填入合适的代码。

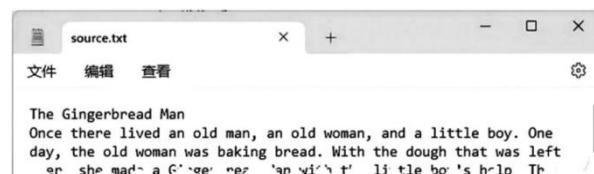
```
m=int(input("路段数量:"))
s=input("坑的坐标依次为:").split(',')
n=len(s)
for i in range(n):
s[i]=int(s[i])
flag=[False] * (n-1)
for i in range(1,m):
k=-1
for j in range(n-1):
if ____①____:
    if k=-1 or s[j+1]-s[j]>s[k+1]-s[k]:
        k=j
flag[k]=True
print("维修管制的路段依次为:")
dis,t=0,0
```

```

for i in range(n-1):
    if flag[i]:
        print(s[t],"~",s[i])
        dis+=s[i]-s[t]+1
        ②
print(s[t],"~",s[n-1])
dis= ③
print("管制总长度为" , dis)

```

15.(2024 届强基联盟联考,13)下列 Python 程序段功能为:统计如图所示英文小说中英文单词出现的次数,且不区分大小字母,输出文章中出现次数最多的英文单词以及出现次数,如有并列则一并输出。



(1)下列程序代码中,加框处的语句____(选填:能/不能)改写成 `elif "a"<=ch<="z"`。

(2)为实现题目所述功能,请为下面划线处填写合适的代码。

```

file=open("source.txt","r",encoding="utf8")
context=""
line=file.readline()
while line:
    context+=line.strip()+" "
    line=file.readline()
    ①
words=
word=""
for i in range(num):
    ch=context[i]

```

```

if "A"<=ch<="Z":
    ch=chr(ord(ch)+32)
if "a"<=ch<="z":
    word+=ch
elif len(word)>0:
    if word in words:
        _____②_____
    else:
        words[word]=1
    word=""
maxcount=0
maxword=[]
for key in words:
    if _____③_____:
        maxcount=words[key]
        maxword=[key]
elif words[key]==maxcount:
    maxword.append(key)
print(maxword,maxcount)

```

16.(2024 届百校起点调研测试,13)某货品仓库最多存储 m 件货物,只能从左右两侧取出,现要在移动其他货物次数尽可能少的情况下取出一件某类货物,取出后,其他货物按照原有顺序重新存入,货物间不留空隙。比如:仓库存储情况列表 $a=["0","A","A","B","B","A","0","0","0"]$,其中"0"表示空位,"A","B"表示位置上货物种类,现需从中取出一件 B 货物,则应从右侧取出,取完后,仓库存储列表更新为 $a=["0","A","A","B","A","0","0","0","0"]$ 。

(1)若要在上述实例中,再取出一件 B 货物,则应从_____ (选填:左侧/右侧)取出。

(2)实现上述功能的部分 Python 程序如下,请在划线处填入合适的代码。

#读取仓库存储情况列表 a,最多存储件数 m,代码略

kind=input("请输入货物:")

left=right=0 #left 记录从左侧取出此货物时需移动其他物品的次数,right 记录
从右侧取出此货物时需移动其他物品的次数

i=0;j=m-1

t=[0,0] #t 表示是否找到此货物

while i<=j and t!= [1,1]:

 if ____①____:

 i=i+1

elif a[i]!=kind:

 left+=1

 i=i+1

else:

 t[0]=1

if a[j]=="0":

 j=j-1

elif a[j]!=kind:

 ____②____

 j=j-1

else:

 t[1]=1

if left>right:

 for k in range(j,m-1):

 a[k]=a[k+1]

a[m-1]="0"

else:

 for k in range(i,0,-1):

 ____③____

a[0]="0"

```
print(a)
```

17.(2024 届发展共同体联考,13)检查数学表达式中的括号是否配对是计算机进行数学计算的重要环节。括号序列“(())”中的“(”与“)”是配对的,而序列“(())(”中的括号则是不配对的。对于不配对的序列,可以将“(”修改为“)”,或者将“)”修改为“(”来实现配对。下图是括号序列“(())(”通过不同的修改方案使其配对所需要的修改次数,最少修改次数为 2。

修改前	修改后	修改次数
(() (())	(((())))	4
() () (()	() (()) ()	2
() () (()	() (() ())	2

请回答下列问题:

- (1)若括号序列为“(())())”,最少需要修改_____次才能使得序列中的括号配对。
- (2)编写程序,计算修改括号序列使其配对的最少次数。部分 Python 程序如下,请在划线处填入合适的代码。

```
s=input() #输入括号序列,序列中仅包含“(” “)”两种字符,且长度为偶数
x=0
ans=0
for i in range(len(s)):
    if s[i]=='(':
        ①
    elif s[i]==')' and x>=1:
        x-=1
    elif s[i]==')' and ②:
        ans+=1;x+=1
    ans+= ③
print(ans)
```

18.(2024 届三校联考,17)

有一小块矿石,质量不超过 200g,现用托盘天平来称重,已知砝码规格有 50g,20g,10g,5g,2g,1g,共 6 种,每种规格的砝码数量均满足需求,游码置于标尺零刻度位置且无需拨动,将该矿石放至托盘天平左盘,不断往右盘加砝码(注:先大后小),直至平衡。根据输入的矿石质量(注:只考虑正整数),输出平衡时所需的各规格砝码数量。

该程序运行的部分界面如下图所示,请在程序划线处填入合适的代码。

请输入矿石质量: 149 所需的砝码: 50g:2个; 20g:2个; 5g:1个; 2g:2个

```
ore=int(input("请输入矿石质量:"))
weights=[50,20,10,5,2,1] #砝码规格
num=[0] * len(weights) #初始化所需的各规格砝码数量
res=_____①_____
i=0
while ore > 0:
    num[i]=ore // weights[i]
    _____②_____
    i+=1
for i in range(len(num)):
    if _____③_____:
        res=res+ str(weights[i])+ "g:"+ str(num[i])+ "个;"
print(res[:-1])
```

19.(2023 台州八校联考,15)近期学校大力宣传信息安全,小张同学想利用近期学习的数据加密知识,对同班同学的密码进行加密,加密规则如下:

- ①大小写转换:将明文中的大写字母改为小写字母。
- ②置换加密:根据输入的置换密码,将字符串中的字母和数字逐个加上置换密码中的对应数字循环左移,置换密码可以循环使用。如原文为“hello2023”,置换密钥为“526”,则加密为“ccfgm6507”。(明文字符 h、e、l、l、o、2、0、2、3 分别往左移

5、2、6、5、2、6、5、2、6 位)。

③除字母、数字以外的字符不变。

(1)若输入的明文为:123Bus,置换密码为:4301,则输出的密文为:_____。

在程序划线处填入合适的代码:

(2)大小写的转换。

```
def change(ch):
```

```
    ch=chr(_____)
```

```
    return ch
```

(3)置换加密。

```
def encrypt(encode,key):
```

```
    q=[]
```

```
    result=" "
```

```
    for x in key:
```

```
        q.append(int(x))#将密钥逐个转为数值,并放到 q 中
```

```
    head=0
```

```
    for ch in encode:
```

```
        if "A"<=ch<="Z":
```

```
            ch=change(ch)
```

```
        if "a"<=ch<="z":
```

```
            ch=chr((ord(ch)-97-q[head])%26+97)
```

```
        elif "0"<=ch<="9":
```

```
            ch=_____①_____
```

```
        head=_____②_____
```

```
        result+=ch
```

```
    return result
```

(4)主程序。

```
mw=input("请输入明文:")
```

```
key=input("请输入置换密钥:")
```

```
ss=encrypt(_____)
```

```
print("密文为:"+ss)
```

20.(2022“七彩阳光”联盟期中,16)某阅卷系统设计如下:

读取学生答案文件后和标准答案进行对照并判分,题型分为单选、多选和填空。

单选阅卷规则:和标准答案一致得全部分数,错选或不选不给分。

多选阅卷规则:和标准答案一致得全部分数,漏选给一半分数,错选或不选不给分。

填空阅卷规则:标准答案表中允许有多种答案存在。学生的答案在标准答案表中存在,得全部分数,不存在或未填不给分。

例如“学生答案.txt”内容如下:

张三|A|D|AC|AD|字符串|

李四|A|D|ACD|A|string|i>=0

标准答案:

题号	1	2	3	4	5	6
题型	单选	单选	多选	多选	填空	填空
答案	A	D	ACD	AD	字符串或 string	i==0 或 i>=0
分值	2	2	2	2	2	2

程序执行结果:

张三成绩:9 分

李四成绩:11 分

(1) 如果学生的答案数据为'王五|A|C|AD|ACD|string|i=0',则输出的结果是_____。

(2)请在横线处填入合适的代码。

```
def dx(s1,s2,n): #对多选题判分
```

```
    flag=True
```

```
    for i in s1:
```

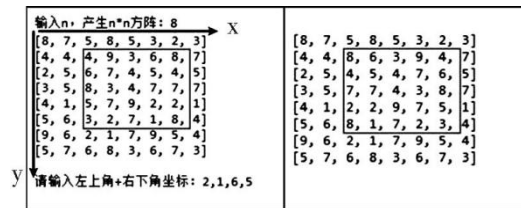
```
        if _____①_____:
```

```

        flag=False
        break
    if s1==s2:
        return n
    elif _____②_____:
        return 0
    else:
        return n/2
f=open('学生答案.txt')
line=f.readline()
#按行读取学生答案.txt 文件
bzda=['单选','A',2,'单选','D',2,'多选','ACD',2,'多选','AD',2,'填空','字符串','string'],2,'
填空','i==0','i>=0'],2]
while line:
    studa=line.split('|') #将字符串以'|'为间隔分割成多个字符串组成的列表
    zf=0
    for i in range(len(studa)-1):
        p=3*i
        if bzda[p]=='单选': #对单选题判分
            if studa[i+1]==bzda[p+1]:
                zf=zf+bzda[p+2]
            elif bzda[p]=='多选': #对多选题判分
                zf=_____③_____
            else:
                if studa[i+1]!=' ' and studa[i+1] in bzda[p+1]:
                    zf=zf+bzda[p+2]
    print(studa[0],'成绩:', zf,'分')
    line=f.readline() #读取下一行
f.close()

```

21.(2022 宁波三锋联盟期中,16)某程序功能如下:输入 n ,生成 $n*n$ 的矩阵,存于列表 a ,内容为 1 至 9 的随机整数。再对以 (x_0,y_0) 和 (x_1,y_1) 为对角顶点的矩形区域中的数据进行水平翻转,并将变换后的二维数组以矩阵形式输出。再输入小于 n 的四个数字(如 a, b, c, d , 四者关系必须满足 $a < c, b < d$),用逗号间隔: 2,1,6,5,则左上角为 $a[1][2]$,右下角为 $a[5][6]$,运行程序后,运行界面如图所示。



(1)以上图中的矩阵为例,如果输入左上角+右下角的坐标为:2,3,7,5。水平翻转后,则元素 $a[3][4]$ 的值在新数组保存在元素 $a[____][____]$ (填写下标)中。

(2)代码如下,请完成填空。

#生成 $n*n$ 的矩阵

import random

$n = \text{int}(\text{input}(\text{'输入 } n, \text{产生 } n*n \text{ 的矩阵:'}))$

$a = [[\text{random.randint}(1,9) \text{ for } i \text{ in range}(n)] \text{ for } j \text{ in range}(n)]$

for i in a:

 print(i)

print()

$x, y = [], []$

$xy = \text{input}(\text{"请输入左上角+右下角坐标:"})$ #输入左上角坐标+右下角坐标,用逗号分隔

$xy = xy + ","$

$c, j = 0, 0$

for i in range(len(xy)):

if $xy[i] == ","$:

if ① :

$x.append(\text{int}(xy[j:i]))$

else:

```

y.append(int(xy[j:i]))
②
c=c+1
for i in range(y[0],y[1]+1,1):
    for j in range(x[0],(x[0]+x[1])/2+1,1):
        a[i][j],a[i][③]=a[i][③],a[i][j]
for i in range(len(a)):
    print(a[i])

```

22.(2022 衢温“5+1”联盟期中,18)某信息仅包含大写字母,字符数不超过 $n*n$ ($n*n$ 为密钥矩阵规模),加密过程如下:

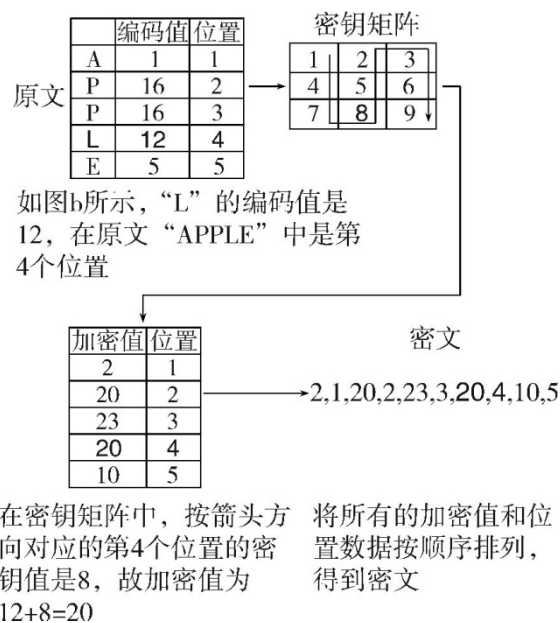


图 a

字母	A	B	C	D	E	F	G	H	I
编码值	1	2	3	4	5	6	7	8	9

字母	J	K	L	M	N	O	P	Q	R
编码值	10	11	12	13	14	15	16	17	18

字母	S	T	U	V	W	X	Y	Z	
编码值	19	20	21	22	23	24	25	26	

图 b

根据上述加密算法编写解密程序。程序在启动时读取密钥矩阵存入列表 a 中,根据程序提示,输入密文,然后显示明文。运行界面如图 c 所示。

请输入密文:2,1,6,2,10,3,12,4,10,5,

原文为:ABCDE

图 c

(1)输入的密文是“8,1,23,2,17,3,33,4,”则原文是_____ (按图 a 中的密钥矩阵)。

(2)实现上述算法的 Python 程序如下,请在划线处填入合适的代码。

```
def getmm(x, n):
    col=(x-1)%n
    row=(x-1)//n
    getmm=a[col*n+row]
    if row % 2==1:
        getmm=_____①_____
    return getmm
n=3
#存储密文数据,偶数位存储加密值,奇数位存储该数据在原文中的位置
#生成规模为 n*n 的密钥矩阵存入列表 a 中
mw=[]
a=[]
for i in range (n*n):
    a.append(i+1)
p=input("请输入密文:") #2,1,6,2,10,3,12,4,10,5,
t=0
k=0
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/488027131142006050>