s[0:4]或 s[:4] 或 s[:-5]得到"1234"　　s[::-1]___得到"987654321"
s[3::-1]_得到"4321"　　s[:4:-1]得到"9876"
s[::-2]或 s[:-10:-2] 或 s[8::-2] 得到"97531"　　s[1:8:2]或 s[1:9:2]得到"2468"

Python 练习 3

课堂练习：　s[起点：终点：方向]　　range(x,x,x)

字符串切片：

s="123456789"

s[0:4]　　得到"1234"
s[3::-1]　　得到"4321"
s[::-2]　　得到"97531"
s[::-1]　　得到"987654321"
s[:4:-1]　　得到"9876"
s[1:8:2]　　得到"2468"

1~5 CDCBD

1. 计算表达式 1/2+2/3+3/5+5/8+…前 30 项的和。实现此功能的 Python 程序如下：

```
a,b=1,2
s=0
for i in range(30):
    [　　]
print(s)
```

方框中的代码由下列 3 个语句组成：①b=b+a；②s=s+a/b；③a=b-a。

下列选项中代码顺序正确的是　（C ）

A. ①③②　　B. ③①②

C. ②①③　　D. ②③①

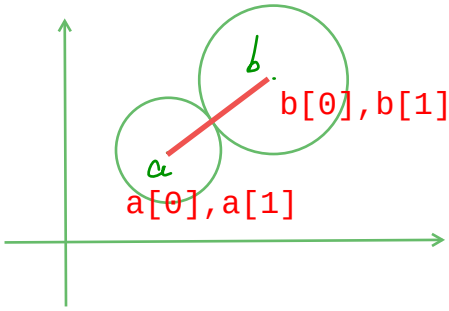
2. (2022 • 衢温 5+1 联考)某张单据上有一个 5 位数的编号 n，如25●●6，其百位数和十位数模糊不清，但是知道这个 5 位数是 23 的倍数。现要设计一个算法，找出所有满足这些条件的 5 位数，并统计个数。现有 Python 程序段如下：

```
c=0
for i in  range(10):
  for j in range(10):
    m=25006+s
    if m%23==0:
      print(m)
      c=c+1
print("满足这些条件的 5 位数总共有:",c,"个" )
```

画线处应填的代码是　（ D ）

A. s=i*10+j　B. s=i*100+j

C. s=i+j*10　D. s=(i*10+j)*10



3. 使用 Python 程序编程探究平面上圆与圆的位置关系，程序代码如下：

```
def judge(a,b):
    dis=(cir[a][0]-cir[b][0])**2+(cir[a][1]-cir[b][1])**2
    if dis==(cir[a][2]+cir[b][2])**2:
        return 1
    return 0
cir={'A':[1,1,1],'B':[4,5,4],'C':[0,2,1]}
#cir 用于存储编号为'A'、'B'、'C'的三个圆的 x,y 坐标及半径信息
cnt=judge('A','B')+judge('A','C')+judge('B','C')
```

运行程序后，变量 cnt 的值是　（C ）

A. 0　B. 1　C. 2　D. 3

4 如下 Python 程序：

```
s1=input("请输入字符串:")
a=[0]*128
for item in s1:
    ch=ord(item)
    a[ch]=a[ch]+1
s2=""
for i in range(len(a)):
    for j in range(a[i]):
        s2=s2+chr(i)
print(s2)
```

ascII表 （128个字符）

0 1 2 ... a b c d..h..n o p q r s t.. u v w x y z

列表 a 各元素的初始值都为 0,s1 中输入的内容为“python21study”。执行该程序后,程序输出的结果中第 6 个字符为 （B ）

12dhnopsttuyy

A.n B.o C.2 D.p

5.(2022 •十校联盟联考)小明想要在数字串 s 中寻找连续数字之和为 k 的子串,若有多个子串符合,则输出第一个子串。例如:s="20220520",k=7,则符合要求的子串为"205"。实现该功能的部分 Python 代码如下:

```
i=0;j=0;sum=0
for  j  in  range(len(s)):
    sum+=int(s[j])
    while  sum>k:
        c=s[i]
        ①
        i+=1
    if  sum==k:
        print("符合要求的子串为:",  ②  )
        break
```

累加合大于指定的 k

画线①②处的语句是 （D ）

A.①sum=sum+int(c) ②s[i:j+1]

B.①sum=sum-int(c) ②s[i-1:j+1]

C.①sum=sum+int(c) ②s[i:j]

D.①sum=sum-int(c) ②s[i:j+1]

6.用火柴棍拼成数字 0~9,具体拼法如下图所示:

(1)编写 Python 程序实现:运行程序时,输入任意两位正整数,输出所需火柴棍的数量。完善下列代码:

```
num=int(input("请输入一个两位正整数:"))
match="6255456376"
a=num%10
b=  ①  num//10  或 (num-a)//10
count=  ②  int(match[a])+int(match[b])
print("共需火柴棍:",count)
```

前者更简洁

没必要再加上str( )

```
count= ② str(int(match[a])+int(match[b]))
print("共需火柴棍:" + count)
```

7.脱氧核糖核酸(DNA)由两条互补的碱基链以双螺旋的方式结合而成。构成 DNA 的碱基共有 4 种,分别为腺嘌呤(A)、鸟嘌呤(G)、胸腺嘧啶(T)和胞嘧啶(C)。在两条互补碱基链的对应位置上,A 总是和 T 配对,G 总是和 C 配对。编写 Python 程序实现如下功能,随机产生一条单链上的碱基序列,输出其对应的互补链上的碱基序列。例如:生成链为 GCTCTTACAT,互补链为 CGAGAATGTA。

(1)实现该功能的程序段如下,请完善程序:

```
import random
DNA=['A','G','T','C']
s=''                          random.choice(['A','G','T','C'] )
for i in range(10):           random.choice("AGTC")
    r=random.choice(DNA)      #随机选取列表 DNA 中的一个元素
    s= ① s+r或r+s
print('生成链:',s)
match={'A':'T','G':'C', 'T':'A','C':'G' }
#定义一个互补碱基对的字典
t=''
for c in s:
    t= ②t+match[c]
print('互补链:',t)
```

等价

错误： t+match["c"] t+match(c)

(2)将程序加框处语句改为 DNA="AGTC",程序是否能正常运行:是 。(选填:是/否)

8 字符串常见操作 4:取子字符后处理。输入一个序列,如果出现连续升序的几个数(或字符),称为一个升序段，升序段字符串长度必须 2 个或 2 个以上字符。如字符串"12312232"中存在 3 个升序段,分别是"123"、"12"和"23",下列程序的功能是统计序列中升序段的个数

```
s=input('输入字符串:')
i=1
k=0
t=0
while i<len(s):
    if s[i-1]<s[i]:
        ①k+=1
    else:
        k=0
    if k==1:
        ②t+=1
    i+=1
print("升序段个数为：",t)
```

9　(2022·9+1 联考)某字符串解密程序运行结果如下图所示。

请输入十六进制密文:9E919697BC 明文为:China

解密算法如下:设十六进制密文的长度为偶数 n,则进行 n/2 轮操作,每轮操作的步骤为:
①从左到右依次取密文中 2 位十六进制数,分别转换为 4 位二进制,不足 4 位的在左边补 0,顺序连接成 8 位二进制(如十六进制 91 转换为二进制为 10010001)。
②将得到的 8 位二进制编码进行取反码(0 变 1,1 变 0)操作,如 10010001 的反码是 01101110。
③将 8 位二进制转化为十进制,并以该十进制作为 ASCII 码,转换为相应的字符。
将每轮得到的字符的倒序连接,即得到最终的明文。
(1)如果输入的密文是“9C9D”,则解密后的明文为 bc 。(小写字母“a”的 ASCII 码为 97)
(2)请在画线处填入合适的代码。

```
def  htob(ch):  #将 1 位十六进制字符转换为 4 位二进制,并将每位取反
    hex={'A':10,'B':11,'C':12,'D':13,'E':14,'F':15}
    if  'A'<=ch<='F':
        ①t=hex[ch]   (不用加int)
    else:
        t=int(ch)
    b=''
    for  i  in  range(4):  为何不用while?
        r=t%2
        b=str(1-r)+b
        t=t//2
    return  b  返回值为字符串
s=input("请输入十六进制密文:")
tmp=''          s:"9C9D"
for  i  in  range(0,len(s),2):  #依次取出两位十六进制密文,并转换为二进制
    t=s[i:i+2]
    ② tmp=tmp+htob(t[0])+htob(t[1])
mw=''   tmp=tmp+htob(s[i])+htob(s[i+1])
for  i  in  range(0,len(tmp),8):  #将每 8 位二进制转换为十进制,并转换为明文字符
    t=tmp[i:i+8]
    x=0
    for  j  in  range(8):
        ③x=x*2+int(t[j])  或  x=x+int(t[j])*2**(8-1-j)
                          或  x=x+int(t[7-j])*2**j
    mw=chr(x)+mw
print("明文为:",mw)
```

9　C　9　D
1001 1100 1001 1101
0110 0011 0110 0010
99　98
c　b

6　3

9E919697BC

结果:tmp:0110 0011 0110 0010

10.将十进制数 n 转二进制数存储在字符变量 s 中

```
s=""
while n>0:
    r=① n % 2
    s=② str(r) + s
    n=③ n//2
print(s)
```

9　C　9　D
1001 1100 1001 1101
0110 0011 0110 0010
6　3　6　2
二进制取反?
相当于十六进制如何运算?