

2023 年转计机考试题与答案（回忆版） --- Python

写在前面：

首先感谢转专业群以及群里同学的热情相助，没有学长学姐的付出转专业不会如此顺利！

我于 23 年申请转计算机，因为机考这一考察形式较新，我们同转计科学院的学子复习时十分迷茫~~（信竞大佬不需要复习）~~。得益于学长学姐的耐心答疑以及前人留下的 22 年回忆版真题（考前复习了一遍，呵，您猜怎么着，原题！），决定以整理原题的方式将善意薪火传递，回馈转群。

如果有所帮助，则是最好不过的事情！

水平有限，还望各位大神斧正，有什么更优解法欢迎指出！

本项目 Github 地址：

https://github.com/CuzWeAre/CUMT_Program-Test

1.给定一个整数，求他的 16 位补码

正整数的补码和该数值的原码（即该数值的二进制形式）相同，例 1 的原码和补码都是 0000000000000001。负数的补码求法是符号位不变，其他位取反，然后结果加 1。例如 -1 的原码是 1000000000000001，补码是 1111111111111111。

```
n = int(input())
if n >= 0:
    n = bin(n)[2:].zfill(16)
    print(n)
else:
    n = abs(n)
    n = list(bin(n)[2:].zfill(15))
    for i in range(len(n)):
        if n[i]=='0':
            n[i]='1'
        else:
            n[i]='0'
    ans = int('1'+''.join(n),2)+1
    print(bin(ans)[2:])
```

2.求 150 以内的 H 半素数

H 数指满足 $4n+1$ ($n \in \mathbb{N}, n \geq 0$) 的数, 例如 1、5、9、13.....

H 素数是只能被 1 和本身整除的数, 例如 5、9、13.....

H 半素数是可以分解为两个 H 素数相乘的数, 例如 25、45、81、85.....

求 150 以内的 H 半素数

```
import itertools

def ishprime(n:int) -> bool: #:int ->bool 可以不写, 说明性的代码
    '''判断是否为 h 素数'''
    if n <=1:
        return False

    #依赖于全局变量 lst_h, 需要在 lst_h 给出后再运行函数。实际上也可以改写函数
    名为 ishprime(n:int,lst_h:list), 把 lst_h 作为参数传递进来
    for i in lst_h[1:]:#把 1 去了
        if n%i == 0 and n!=i:
            return False
    return True

lst_h = [4*n+1 for n in range(150) if 4*n+1<150] #range(150)是个随便的
值, 实际上只要 4*n+1 能取到最接近 150 的值范围任意
lst_hprime = [x for x in lst_h if ishprime(x)]
hprime_combinations = itertools.combinations(lst_hprime,2)#翘课了, 这个库
蛮好用的建议学。递归造轮子太折磨 TAT
#print(list(hprime_combinations)) #去掉注释执行可观察返回的样子
#print(hprime_combinations) #和上面比较一下有什么不同
lst_halfprime = [[k[0]*k[1],k] for k in hprime_combinations if
k[0]*k[1]<150]
lst_halfprime += [[k**2,(k,k)] for k in lst_hprime if k**2<150]#加上
k**2 排序方便
lst_halfprime.sort()
for i in lst_halfprime:
    print(f'{i[0]}={i[1][0]}*{i[1][1]}')
```

3.求黑洞数

读取一个四位不全相同的数字, 这四个数字排列组合出最大的数和最小的数做差, 之后重复这个步骤, 最多 7 次数字最终便会变成 6174。

例如 1234 这个数字只需要 2 次就会掉入黑洞。

现在要求你编写程序, 从键盘读入一个四位不全相同的数字, 求需要像上面那样处理多少次掉入黑洞。

```
N = list(input())
for i in range(8):
    #最多 7 次掉入黑洞
    N_min = int(''.join(sorted(N)))
    N_max = int(''.join(sorted(N,reverse=True)))
    N = N_max-N_min
    if N == 6174:
        print(i)
        break
    N = list(str(N))
```

4.求回文日期

回文日期指的是一个日期表示为 `yyyymmdd` 的形式后, `yyyymmdd` 是个回文数。例如 20200202 是一个回文日期, 它的下一个回文日期是 20211202。

给定一个 `yyyymmdd` 的日期, 求下一个回文日期是在哪一天。

```
#ChatGpt 提供的答案
#鼠鼠我啊, 机考没做出来, 疯狂造轮子, 考得是一点不会(T▽T)
from datetime import datetime, timedelta

def is_palindrome(num):
    """判断给定的整数是否为回文数"""
    str_num = str(num)
    return str_num == str_num[::-1]

def next_palindrome_date(date):
    """计算给定日期的下一个回文日期"""
    curr_date = datetime.strptime(date, "%Y%m%d")
    # 逐个增加日期, 直到找到回文日期为止
    while True:
        curr_date += timedelta(days=1)
        next_date_str = datetime.strftime(curr_date, "%Y%m%d")
        if is_palindrome(int(next_date_str)):
            return next_date_str

date_str = input()#输入一个日期 (格式为 YYYYMMDD)
datetime.strptime(date_str, "%Y%m%d")
next_palindrome = next_palindrome_date(date_str)
print(next_palindrome)

#不用库的版本
def is_palindrome(num):
    """判断给定的整数是否为回文数"""
    str_num = str(num)
    return str_num == str_num[::-1]

def is_valid_date(date):
    """判断给定的日期是否合法"""
    year, month, day = int(date[:4]), int(date[4:6]), int(date[6:])
    if year < 1000 or year > 9999 or month < 1 or month > 12:
        return False
```

```

    if month in {1, 3, 5, 7, 8, 10, 12}:
        return day >= 1 and day <= 31
    elif month in {4, 6, 9, 11}:
        return day >= 1 and day <= 30
    elif year % 4 == 0 and (year % 100 != 0 or year % 400 == 0):
        return day >= 1 and day <= 29
    else:
        return day >= 1 and day <= 28

def next_palindrome_date(date):
    """计算给定日期的下一个回文日期"""
    year, month, day = int(date[:4]), int(date[4:6]), int(date[6:])
    while True:
        # 逐个增加日期
        day += 1
        if day > 31 or (day > 30 and month in {4, 6, 9, 11}) or (day >
29 and month == 2) or (day > 28 and month == 2 and (year % 4 != 0 or
(year % 100 == 0 and year % 400 != 0))):
            day = 1
            month += 1
        if month > 12:
            month = 1
            year += 1
        # 构造下一个日期并判断是否为回文日期
        next_date = f"{year:04d}{month:02d}{day:02d}"
        if is_valid_date(next_date) and is_palindrome(next_date):
            return next_date

```

5.填写矩阵

比 22 年机考题简单，类似题目可看 leetcode.cn 54,59 题 关键词：螺旋矩阵
键盘读入一个正偶数 n ，生成一个 $(n-1)*(n-1)$ 的矩阵，从外到里递增，即最里面的数是 $n/2$ 。

例：输入 10

```
1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1
```

```
n = int(input())
matrix = [[0 for _ in range(n-1)] for _ in range(n-1)] #_代表不需用到迭代
数具体值
for i in range(n-1):
    for up in range(i,n-1-i):
        matrix[i][up]=i+1
    for down in range(i,n-1-i):
        matrix[-i-1][down]=i+1
    for left in range(i,n-1-i):
        matrix[left][i]=i+1
    for right in range(i,n-1-i):
        matrix[right][-i-1]=i+1

for x in matrix:
    x = map(str,x)
    print(' '.join(x))

#可以把上面的循环注释掉，下面的代码去掉注释，以便于观察每次操作后矩阵的变
化
# for x in matrix:
#     x = map(str,x)
#     print(' '.join(x))

# print()
```