

Python实验5

原创

今天你学习了么?? 于 2020-11-14 20:24:22 发布 3627 收藏 56

文章标签: [python 算法](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_52531012/article/details/109690943

版权

实验五 常用的算法设计方法

有更好的程序请留言哦, 可以一起交流

1.从键盘输入N的值, 求

$$X = \sum_{i=1}^N (i^4 - 2i^2 + 1)$$

```
N=int(input())
sum1=sum2=0
i=1
for i1 in range (1,N+1):
    if i1<=N:
        sum1+=(i1-pow(i1,4))
        i1
print(sum1)
for i2 in range (2,N+1,2):
    if i2<=N:
        sum2+=((i2)/2-pow(i2,6))
        i2
print(sum2)
```

结果演示:

```
5
-4291728792
-4157.0

6
-4479117396
-50810.0
```

2.设 $s = \sum_{i=1}^n \frac{1}{i}$

```
n=int(input("请输入n值: "))
s=0
for i in range (1,n+1):
    s+=1/i
    i+1
print(float(s))
```

结果演示:

```
请输入n值: 1673
7.999888200430668
```

3. [1,100]间有奇数个不同因子的整数共有多少个？其中最大的一个是什么数？

```
m=0
for i in range(1,101):
    n=0
    for j in range(1,i):
        if i%j==0:
            n+=1
    if n%2!=0:
        max=i
        m+=1
    print("{0}的因子个数为{1}.".format(i,n))
print(m,max)
```

结果演示：

```
省略了部分结果
2的因子个数为1.
3的因子个数为1.
5的因子个数为1.
...
96的因子个数为11.
97的因子个数为1.
98的因子个数为5.
99的因子个数为5.
90 99
```

4.梅森尼数是指 $2^n - 1$ 为素数的数n，求[1, 21]内有多少个梅森尼数以及最大的梅森尼数。

```
def sushu(n):
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
def meiNumber(n):
    list = []
    for i in range(2, n):
        if sushu(i):
            if sushu(2**i-1):
                list.append(2**i-1)
    return list
print (meiNumber(21))
print("最大的梅森尼数为：",max(meiNumber(21)))
```

结果演示：

```
[3, 7, 31, 127, 8191, 131071, 524287]
最大的梅森尼数为： 524287
```

5.已知 $A>B>C$,且 $A+B+C<100$,求满足

```
for A in range(1,100):
    for B in range(1, 100):
        for C in range(1, 100):
            if A>B>C:
                if A+B+C<100:
                    if 1/(A*A)+1/(B*B)==1/(C*C):
                        print(A,B,C)
```

结果演示:

```
20 15 12
40 30 24
```

6. 已知 $f(t) = \sqrt{\cos(2t) + 4\sin(2t) + 5}$, 利用矩形法和梯形法求

$\int_0^{2\pi}$

```
#矩形法
import math

n=100      #分成100份
a=0        #上限
b=2*math.pi #下限
s=0        #面积
d=0        #边长

w=(b-a)/n
for i in range(1,101):
    d=i*w
    s+=math.sqrt(math.cos(d) + 4 * math.sin(2 * d) + 5)*w
print(s)

#梯形法
import math

n=100      #分成100份
a=0        #上限
b=2*math.pi #下限
s=0        #面积
d=0        #边长

w=(b-a)/n
for i in range(1,101):
    d=i*w      #右横坐标
    D=(i-1)*w #左横坐标
    s+=((math.sqrt(math.cos(d) + 4 * math.sin(2 * d) + 5))+(math.sqrt(math.cos(D) + 4 * math.sin(2 * D) + 5)))*w/2
print(s)
```

结果演示:

```
13.261247839372379
13.261247839372386
```

7. 用牛顿迭代法求方程

$e^{-x} - x = 0$ 在 $x = -2$ 附近的一个实根。直到满足 $|x_{n+1} - x_n| < 1e-6$

```
import math
def f(x):
    return(math.exp(-x)-x)
def g(x):
    return(-(math.exp(-x))-1)

x=-2
while (1):
    a=x
    x=x-f(x)/g(x)
    if (abs(x-a)<(1e-6)):
        break
print(x)
```

结果演示:

```
0.5671432904097837
```

8.设

$f(x) = x^3 - a$ 用牛顿迭代法推导方程 $f(x) = 0$ 的根的迭代公式。显然方程的根为

```
a=int(input("请输入a值: "))
```

```
def f(x):  
    return(x**3-a)
```

```
def g(x):  
    return(3*(x**2))
```

```
x=a  
while (1):  
    b=x  
    x=x-f(x)/g(x)  
    if (abs(x-b)<(1e-5)):  
        break
```

```
print(x)
```

结果演示:

```
请输入a值: 27
```

```
3.0
```



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)