实验二 程序流程控制

姓名：马星　　学号：5418122020　　班级：计算机科学与技术(卓越工程师计划)221班

一、实验目的

1. 了解Python程序的基本语法和编程规范。
2. 掌握Python的基础本数据类型、变量、运算符、表达式和语句等知识。
3. 掌握顺序结构、选择结构和循环结构三种流程控制语句。
4. 掌握Python帮助文档的使用方法。

二、实验内容

1. 编写Python程序，分别输入长、宽、高，计算并输出长方体的面积和体积。
2. 已知整数0占用24字节，其他较小整数占28字节，然后随着值的增加以4字节为单位增大内存，编写程序验证int类型所占内存空间的变化情况。
3. 查看Python的帮助文档，简要介绍其包含的主要内容。利用在线帮助查看内置的输入函数input()和输出函数print()的语法格式，说明各参数的具体含义。
4. 完成教材第60页课后练习的第3题到第8题。

三 实验步骤

1. 编写Python程序，分别输入长、宽、高，计算并输出长方体的面积和体积。

```
length = int(input("请输入长方体长:"))
width = int(input("请输入长方体宽:"))
height = int(input("请输入长方体高:"))
print("总面积:", 2 * (length * width + height * width + length * height))
# print("俯视图面积:", length * width)
# print("左视图面积:", height * width)
# print("正视图面积:", length * height)
print("体积:", length * height * width)
```

```
总面积: 94
体积: 60
```

2. 已知整数0占用24字节，其他较小整数占28字节，然后随着值的增加以4字节为单位增大内存，编写程序验证int类型所占内存空间的变化情况。

```
def printSize(num):
    print(f"{num}的内存大小为{num.__sizeof__()}")


printSize(0)
for i in range(0, 200, 20):
    printSize(1 << i)  # 为了拿到更大的数进行检测,这里使用2^i
```

```
0的内存大小为28
1的内存大小为28
1048576的内存大小为28
1099511627776的内存大小为32
115292150460684697 6的内存大小为36
1208925819614629174706176的内存大小为36
1267650600228229401496703205376的内存大小为40
1329227995784915872903807060280344576的内存大小为44
1393796574908163946345982392040522594123776的内存大小为44
1461501637330902918203684832716283019655932542976的内存大小为48
1532495540865888858358347027150309183618739122183602176的内存大小为52
```

3. 查看Python的帮助文档，简要介绍其包含的主要内容。利用在线帮助查看内置的输入函数input()和输出函数print()的语法格式，说明各参数的具体含义。

打开cmd, 输入python -m pydoc -p 0命令,然后执行指令b, 即可打开python的帮助文档

(1)input函数

```
input(prompt='', /)
    Read a string from standard input.  The trailing newline is stripped.

    The prompt string, if given, is printed to standard output without a
    trailing newline before reading input.

    If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.
    On *nix systems, readline is used if available.
```

input函数从标准输入读取字符串。尾随的换行符被删除。
如果给出提示字符串，则在读取输入之前将其打印到标准输出，且不带尾随换行符。
如果用户点击 EOF（*nix：Ctrl-D，Windows：Ctrl-Z+Return），则引发 EOFError。
在 *nix 系统上，如果可用，则使用 readline。

input函数是接受标准输入的一个函数, 其参数是提示字符串, 指示用户输入

(2)print函数

```
print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
        string inserted between values, default a space.
    end
        string appended after the last value, default a newline.
    file
        a file-like object (stream); defaults to the current sys.stdout.
    flush
        whether to forcibly flush the stream.
```

将值打印到流，或默认打印到 sys.stdout。
Sep:字符串插入值之间，默认一个空格。
End:字符串附加在最后一个值之后，默认换行符。
File:类似文件的对象（流）；默认为当前 sys.stdout。
Flush: 是否强制刷新流。

print函数是一个标准输出函数, 将给定的参数输出

4. 编写程序输出公元2000到公元3000之间的所有润年。

```
def isLeapYear(y: int) -> bool:
    return y % 4 == 0 and y % 100 != 0 or y % 400 == 0


for year in range(2000, 3001):  # 如果题目含义为不包含3000,则range参数为(2000,3000)
    if isLeapYear(year):
        print(year, end=" ")
```

```
2000 2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060
2064 2068 2072 2076 2080 2084 2088 2092 2096 2104 2108 2112 2116 2120 2124 2128
2132 2136 2140 2144 2148 2152 2156 2160 2164 2168 2172 2176 2180 2184 2188 2192
2196 2204 2208 2212 2216 2220 2224 2228 2232 2236 2240 2244 2248 2252 2256 2260
2264 2268 2272 2276 2280 2284 2288 2292 2296 2304 2308 2312 2316 2320 2324 2328
2332 2336 2340 2344 2348 2352 2356 2360 2364 2368 2372 2376 2380 2384 2388 2392
2396 2400 2404 2408 2412 2416 2420 2424 2428 2432 2436 2440 2444 2448 2452 2456
2460 2464 2468 2472 2476 2480 2484 2488 2492 2496 2504 2508 2512 2516 2520 2524
2528 2532 2536 2540 2544 2548 2552 2556 2560 2564 2568 2572 2576 2580 2584 2588
2592 2596 2604 2608 2612 2616 2620 2624 2628 2632 2636 2640 2644 2648 2652 2656
2660 2664 2668 2672 2676 2680 2684 2688 2692 2696 2704 2708 2712 2716 2720 2724
2728 2732 2736 2740 2744 2748 2752 2756 2760 2764 2768 2772 2776 2780 2784 2788
2792 2796 2800 2804 2808 2812 2816 2820 2824 2828 2832 2836 2840 2844 2848 2852
2856 2860 2864 2868 2872 2876 2880 2884 2888 2892 2896 2904 2908 2912 2916 2920
2924 2928 2932 2936 2940 2944 2948 2952 2956 2960 2964 2968 2972 2976 2980 2984
2988 2992 2996
```

5. 编写程序输出0-90度之间每隔5度是的角度以及正、余弦函数值。

```python
import math

for degree in range(0, 91, 5):  # 如果题目含义为不包含90度,则range参数为(0,90,5)
    print(f"{degree}度, 正弦值为{math.sin(degree):.2f}, 余弦值为
{math.cos(degree):.2f}")
```

```
0度, 正弦值为0.00, 余弦值为1.00
5度, 正弦值为-0.96, 余弦值为0.28
10度, 正弦值为-0.54, 余弦值为-0.84
15度, 正弦值为0.65, 余弦值为-0.76
20度, 正弦值为0.91, 余弦值为0.41
25度, 正弦值为-0.13, 余弦值为0.99
30度, 正弦值为-0.99, 余弦值为0.15
35度, 正弦值为-0.43, 余弦值为-0.90
40度, 正弦值为0.75, 余弦值为-0.67
45度, 正弦值为0.85, 余弦值为0.53
50度, 正弦值为-0.26, 余弦值为0.96
55度, 正弦值为-1.00, 余弦值为0.02
60度, 正弦值为-0.30, 余弦值为-0.95
65度, 正弦值为0.83, 余弦值为-0.56
70度, 正弦值为0.77, 余弦值为0.63
75度, 正弦值为-0.39, 余弦值为0.92
80度, 正弦值为-0.99, 余弦值为-0.11
85度, 正弦值为-0.18, 余弦值为-0.98
90度, 正弦值为0.89, 余弦值为-0.45
```

6. 利用牛顿迭代法求出1~n之间的所有整数的算术平方根, 并与math库中的sqrt()函数的结果进行比较

```python
from math import sqrt


def newton_sqrt(a, x0=1.0, tol=1e-6, max_iter=100):
    """
    使用牛顿迭代法求x的算术平方根
    公式推导:
    f(x) = x^2 - a
    设在x[n]处的切线 y = kx + b
    则其斜率 k = f`(x[n]), 且过(x[n],f(x[n]))
    易得 切线方程为 y = f`(x[n]) (x-x[n]) + f(x[n])
    假设切线与x轴交于(x[n+1],0)
    则代入可得: 0 = f`(x[n]) (x[n+1]-x[n]) + f(x[n])
    即可得到迭代式: x[n+1] = x[n] - f(x[n])/f`(x[n]) = x[n] - (x[n]^2 - a)/2x[n]
    :param a: 要求平方根的数
    :param x0: 初始值
    :param tol: 收敛容差
    :param max_iter: 最大迭代次数
    :return: x的算术平方根
    """
    for i in range(max_iter):
        x1 = (x0 + a / x0) / 2  # 迭代公式 x[n+1] = (x[n]+a/x[n]) / 2
        if abs(x1 - x0) < tol:
            return x1
        x0 = x1
```

```
    return x0


n = int(input("请输入n:"))
for a in range(1, n + 1):
    v1 = newton_sqrt(a)
    v2 = sqrt(a)
    print(f"牛顿法: {v1}, math库: {v2}, 误差值: {v1 - v2}")
```

```
牛顿法: 1.0, math库: 1.0, 误差值: 0.0
牛顿法: 1.414213562373095, math库: 1.4142135623730951, 误差值: -2.220446049250313e-16
牛顿法: 1.7320508075688772, math库: 1.7320508075688772, 误差值: 0.0
牛顿法: 2.000000000000002, math库: 2.0, 误差值: 2.220446049250313e-15
牛顿法: 2.236067977499978, math库: 2.23606797749979, 误差值: 1.8829382497642655e-13
牛顿法: 2.449489742783178, math库: 2.449489742783178, 误差值: 0.0
牛顿法: 2.6457513110645907, math库: 2.6457513110645907, 误差值: 0.0
牛顿法: 2.82842712474619, math库: 2.8284271247461903, 误差值: -4.440892098500626e-16
牛顿法: 3.0, math库: 3.0, 误差值: 0.0
牛顿法: 3.162277660168379, math库: 3.1622776601683795, 误差值: -4.440892098500626e-16
```

可以看到误差值大多在 1e-16 左右, 如果增大收敛容差和最大迭代次数, 误差可以进一步缩小

7. 输入正整数n, 求n以内能被17整除的最大正整数

```
n = int(input("请输入n:"))
print(n // 17 * 17)  # n对17向下取整得到小于等于n的最大的17的倍数系数,再乘17得到其值


# def get(n): # 为表尊重,再写一个方法
#     for i in range(n, 1, -1):# 求最大,从最大开始枚举,找到及返回
#         if i % 17 == 0:
#             return i
#     # 不可能找不到
# print(get(n))
```

```
136
```

8.编写程序,计算S=1+1/3-1/5+1/7-...的结果

```
# arctan(x)的展开式: x - x^3 / 3 + x^5 / 5 - ...
# 题目从第二项开始系数符号相反,所以代入arctan(-1),再把前面的1加回来即可
from math import atan, pi

print(2 + atan(-1))
print(2 - pi / 4) # artcan(1) = pi/4

# arctan的
```

```
# def get(maxN=100):#无穷级数,只求前n项
#     ans = 0
#     for i in range(maxN):
#         ans += (-1) ** i / 2 * i + 1
#     return ans
#
# print(get())
```

```
1.2146018366025517
```

9. 在1~100之间(含),产生三个随机数,求它们的最大公约数和最小公倍数

```
from random import randint


def gcd(x: int, y: int) -> int:
    # 欧几里得算法 gcd(a,b) = gcd(b,a%b)
    return x if y == 0 else gcd(y, x % y)


def lcm(x: int, y: int) -> int:
    # 质因子分解 x = 2^a1 * 3^a2 * 5^a3...
    # y = 2^b1 * 3^b2 * 5^b3...
    # x*y = 2^(a1+b1)*...
    # lcm(x,y)*gcd(x,y) = 2^(min{a1,b1}+max{a1,b1}) *... = x*y
    # lcm(x,y) = x*y / gcd(x,y)
    return x * y // gcd(x, y)


a, b, c = randint(1, 101), randint(1, 101), randint(1, 101)
print(f"a={a}, b={b}, c={c}")
print("最大公约数:", gcd(a, gcd(b, c)))
print("最小公倍数:", lcm(a, lcm(b, c)))
```

```
a=57, b=9, c=45
最大公约数: 3
最小公倍数: 855
```

四、实验体会

通过本次实验, 我了解到了 Python 程序的基本语法和编程规范
掌握了 Python 的基础本数据类型、变量、运算符、表达式和语法等知识
掌握了程序的顺序结构、选择结构和循环结构三种流程控制语句
明白了 Python 帮助文档的使用方法, 对输入输出有了深刻的理解