

第一次实验课作业

姓名：林正阅 学号：PB24051079

Part I

计算等额本金还款金额

1 问题描述

1. 计算等额本息还款金额



人工智能
数学原理与算法

等额本息是一种分期偿还贷款的方式，即借款人每月按相等的金额偿还贷款本息，每月还款金额 P 可根据贷款总额 A 、年利率 r 和贷款月数 n 计算得到，公式为

$$P = \frac{\frac{r}{12}A}{1 - (1 + \frac{r}{12})^{-n}}$$

计算当贷款金额为 1000000，贷款时间为 30 年，年利率分别为 4%、5% 和 6% 时的每月还款金额和还款总额。输出结果在小数点以后保留两位有效数字。

答案：

(4774.15, 1718695.06)
(5368.22, 1932557.84)
(5995.51, 2158381.89)

图 1

2 问题分析

2.1 问题分析

等额本息是一种常见的分期偿还贷款方式，即借款人每月按相等的金额偿还贷款本息。每月还款金额 P 可根据贷款总额 A 、年利率 r 和贷款月数 n 计算得到。

2.1.1 计算公式

每月还款金额 P 的计算公式为：

$$P = \frac{\frac{r}{12} \cdot A}{1 - (1 + \frac{r}{12})^{-n}}$$

其中：

- A 为贷款总额

- r 为年利率
- n 为贷款月数
- P 为每月还款金额

2.1.2 计算参数

在本问题中，具体参数如下：

- 贷款总额： $A = 1,000,000$ 元
- 贷款期限：30 年，即 $n = 30 \times 12 = 360$ 个月
- 年利率：分别考虑 $r = 4\%, 5\%, 6\%$ 三种情况

2.1.3 计算目标

1. 计算三种年利率下的每月还款金额 P
2. 计算还款总额：总还款额 $= P \times n$
3. 所有结果保留两位小数

3 结果展示

以下是计算等额本息还款额的 Python 代码实现：

```

1 price=1000000
2 year=30
3
4 def p(ratio):
5     pay=(price*ratio/12)/(1-(1+ratio/12)**(-12*year))
6     sum=pay*12*year
7     return (round(pay, 2), round(sum, 2))
8
9 print(p(0.04))
10 print(p(0.05))
11 print(p(0.06))

```

结果为：

(4774.15, 1718695.06)
 (5368.22, 1932557.84)
 (5995.51, 2158381.89)

Part II

计算三角形的内角

1 问题描述

2. 计算三角形的内角



人工智能
数学原理与算法

定义三个变量 “a=3; b=6; c=7” 表示一个三角形的三个边的长度，使用公式

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma$$

分别计算三个内角 (α, β, γ) 的度数，然后检验等式 $\alpha + \beta + \gamma = 180$ 是否成立。

答案：

25.208765296758365 58.41186449479884 96.37937020844281

图 2

2 问题分析

2.1 问题分析

已知三角形三条边的长度，计算其三个内角的度数，并验证三角形内角和定理。

2.1.1 计算公式

根据余弦定理，三角形内角的计算公式为：

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma$$

其中：

- a,b,c 分别为三角形的三条边长
- α, β, γ 分别为对应边 a,b,c 的对角
- 计算得到余弦值后，使用反余弦函数得到角度值（单位为度）

2.1.2 计算参数

在本问题中，具体参数如下：

- 边长：a=3, b=6, c=7
- 角度单位：度 ($^{\circ}$)

2.1.3 计算目标

1. 计算三个内角 α, β, γ 的度数
2. 验证三角形内角和定理: $\alpha + \beta + \gamma = 180$
3. 所有结果保留两位小数

3 结果展示

以下是计算三角形内角的 Python 代码实现:

```
1 import math
2
3 a=3
4 b=6
5 c=7
6
7 x=math.degrees(math.acos((a**2-b**2-c**2)/(-2*c*b)))
8 y=math.degrees(math.acos((b**2-a**2-c**2)/(-2*c*a)))
9 z=math.degrees(math.acos((c**2-b**2-a**2)/(-2*a*b)))
10
11 print(x,y,z)
12 print(x+y+z)
```

结果为:

```
25.208765296758365 58.411864494798834 96.37937020844281
180.0
```

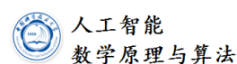
三个角度相加等于 180 度

Part III

list 类型的运算

1 问题描述

3. list 类型的运算



定义两个列表 “s=[2,4,0,1,3,9,5,8,6,7]; t=[2,6,8,4]”。在 lpython 中对于 s 和 t 运行程序 2.9 中 “In[30]:” 行开始的所有运算, 并记录输出结果。若某个运算修改了 s, 在运行下一个运算之前需要再设置 s=[2,4,0,1,3,9,5,8,6,7]。若某个运算需要一些参数 (如 i, j 和 x 等), 可自行设定。

图 3

2 问题分析

对原代码更改 s 的取值，记录输出结果

3 结果展示

以下是记录的输出结果：

```
1 In[29]: s=[2,4,0,1,3,9,5,8,6,7] t=[2,6,8,4]
2 In[30]: s[3] = 8; s
3 Out[30]: [2, 4, 0, 8, 3, 9, 5, 8, 6, 7]
4 In[31]: s=[2,4,0,1,3,9,5,8,6,7]; s[1:4] = t; s
5 Out[31]: [2, 2, 6, 8, 4, 3, 9, 5, 8, 6, 7]
6 In[32]: s=[2,4,0,1,3,9,5,8,6,7]; del s[2:4]; s
7 Out[32]: [2, 4, 3, 9, 5, 8, 6, 7]
8 In[23]: s=[2,4,0,1,3,9,5,8,6,7]; s[0:7:2] = t; s #为了使数量对等，将5改为7
9 Out[33]: [2, 4, 6, 1, 8, 9, 4, 8, 6, 7]
10 In[34]: s=[2,4,0,1,3,9,5,8,6,7]; del s[0:5:2]; s
11 Out[34]: [4, 1, 9, 5, 8, 6, 7]
12 In[35]: s=[2,4,0,1,3,9,5,8,6,7]; s.clear(); s
13 Out[35]: []
14 In[36]: s=[2,4,0,1,3,9,5,8,6,7]; s.copy()
15 Out[36]: [2, 4, 0, 1, 3, 9, 5, 8, 6, 7]
16 In[37]: s=[2,4,0,1,3,9,5,8,6,7]; s.extend(t); s
17 Out[37]: [2, 4, 0, 1, 3, 9, 5, 8, 6, 7, 2, 6, 8, 4]
18 In[38]: s=[2,4,0,1,3,9,5,8,6,7]; s+=t; s
19 Out[38]: [2, 4, 0, 1, 3, 9, 5, 8, 6, 7, 2, 6, 8, 4]
20 In[39]: s=[2,4,0,1,3,9,5,8,6,7]; s*=3; s
21 Out[39]: [2,4,0,1,3,9,5,8,6,7,2,4,0,1,3,9,5,8,6,7,2,4,0,1,3,9,5,8,6,7]
22 In[40]: s=[2,4,0,1,3,9,5,8,6,7]; s.insert(2, 19); s
23 Out[40]: [2, 4, 19, 0, 1, 3, 9, 5, 8, 6, 7]
24 In[41]: s=[2,4,0,1,3,9,5,8,6,7]; (s.pop(2),s)
25 Out[41]: (0, [2, 4, 1, 3, 9, 5, 8, 6, 7])
26 In[42]: s=[2,4,0,1,3,9,5,8,6,7]; (s.remove(7), s)#由于s中不存在11，将其换为7
27 Out[42]: (None, [2, 4, 0, 1, 3, 9, 5, 8, 6])
28 In[43]: s=[2,4,0,1,3,9,5,8,6,7]; s.reverse(); s
29 Out[43]: [7, 6, 8, 5, 9, 3, 1, 0, 4, 2]
30 In[44]: s=[2,4,0,1,3,9,5,8,6,7]; s.sort(); s
31 Out[44]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
32 In[45]: s=[2,4,0,1,3,9,5,8,6,7]; s.sort(reverse=True); s
33 Out[45]: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
34 ##### end of Listing 2.9 #####
```