Introduction to computing C

2024-2025 Fall Midterm Mini-lecture
“信心行动” 期中学业辅导活动

Something about ME!

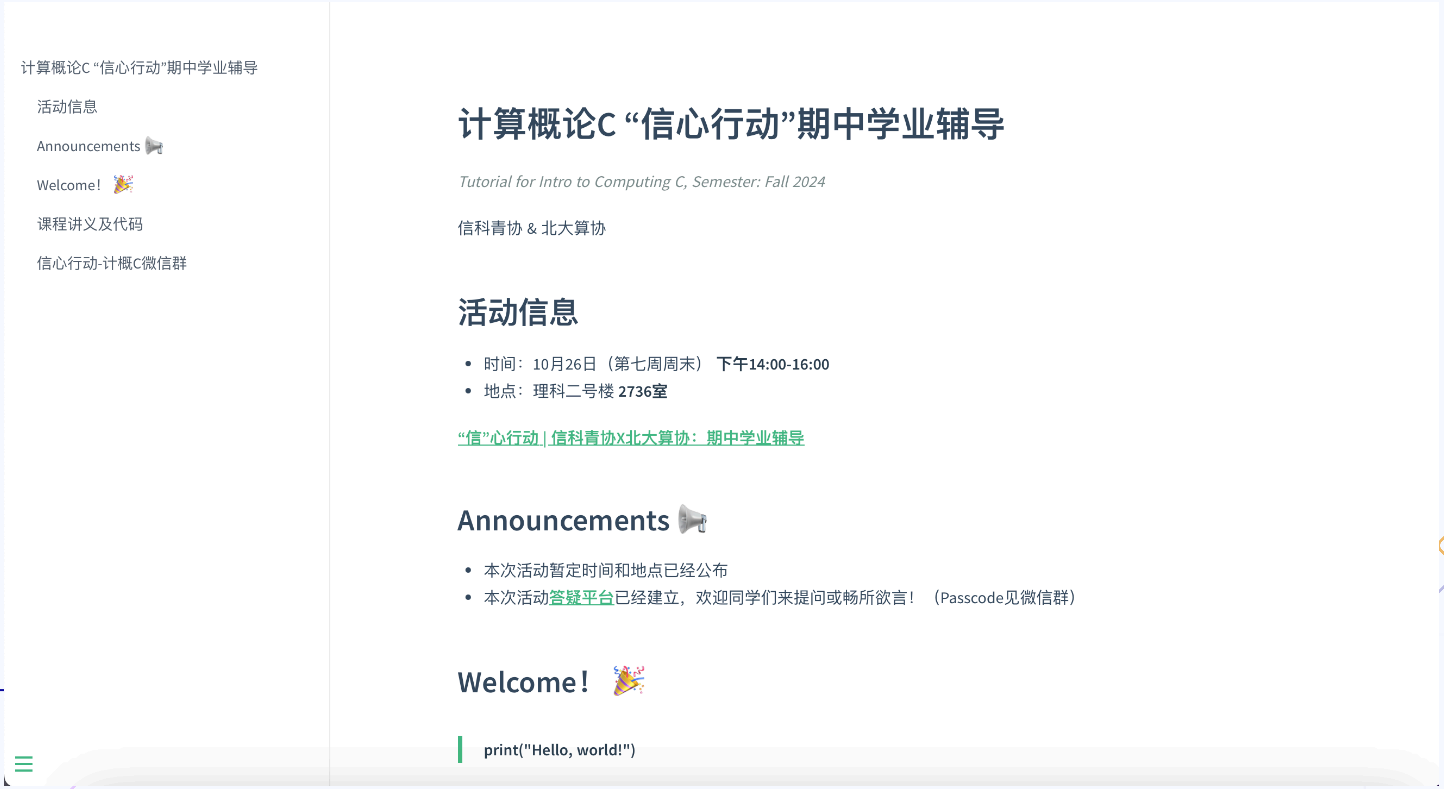
曹彧 · Calvin Cao

信息科学技术学院 2023级 本科生

（高中无竞赛经验和编程基础）

Something about The Activity!

Web



计算概论C “信心行动”期中学业辅导

活动信息

Announcements 📢

Welcome！ 🎉

课程讲义及代码

信心行动-计概C微信群

计算概论C “信心行动”期中学业辅导

Tutorial for Intro to Computing C, Semester: Fall 2024

信科青协 & 北大算协

活动信息

- 时间：10月26日（第七周周末） **下午14:00-16:00**
- 地点：理科二号楼 **2736室**

“信”心行动 | 信科青协X北大算协：期中学业辅导

Announcements 📢

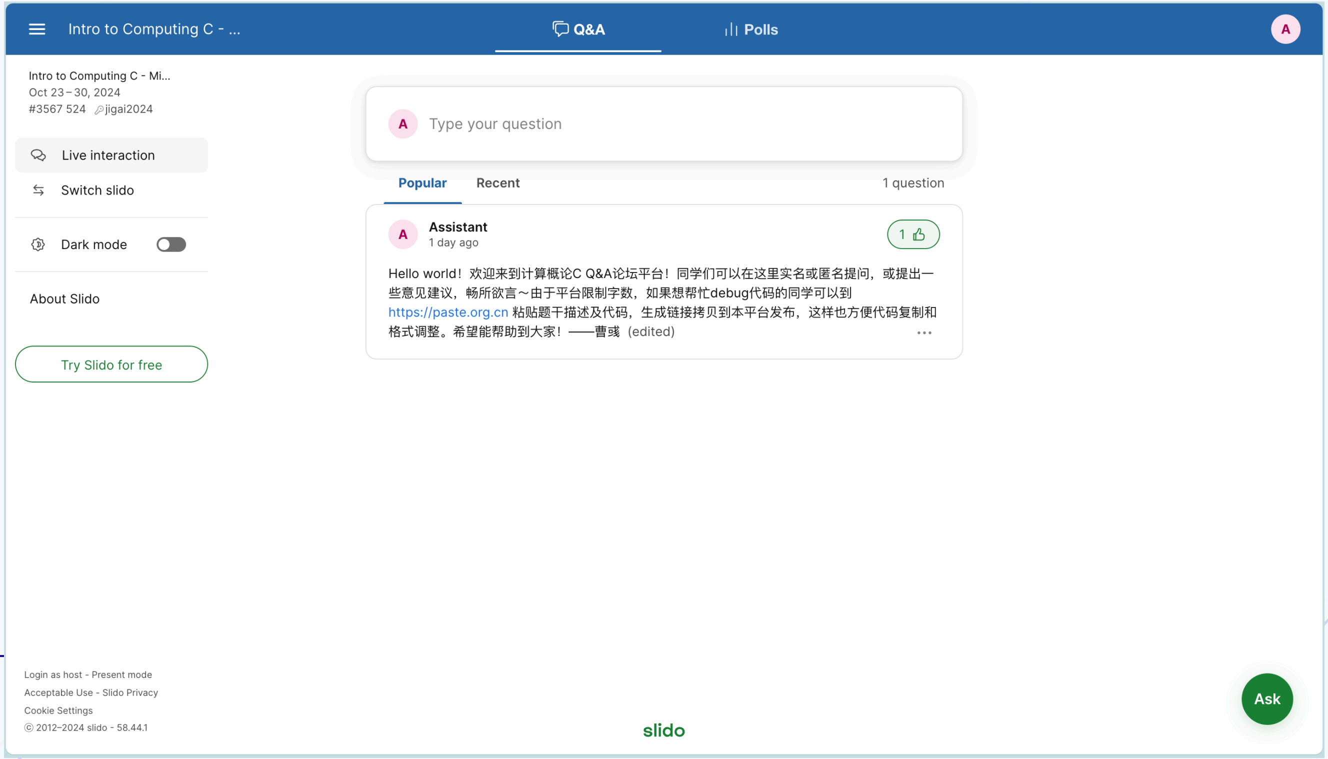
- 本次活动暂定时间和地点已经公布
- 本次活动答疑平台已经建立，欢迎同学们来提问或畅所欲言！ （Passcode见微信群）

Welcome！ 🎉

> print("Hello, world!")

Something about The Activity!

Q&A



Intro to Computing C - Mi...
Oct 23 – 30, 2024
#3567 524 jigai2024

Assistant
1 day ago
1

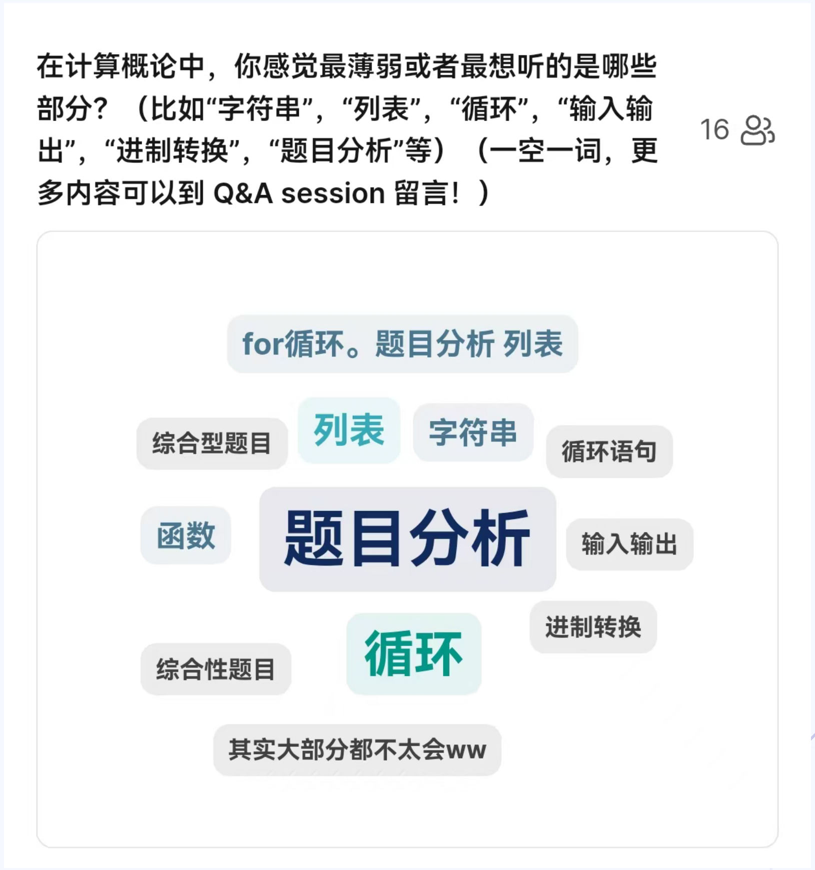
Hello world！欢迎来到计算概论C Q&A论坛平台！同学们可以在这里实名或匿名提问，或提出一些意见建议，畅所欲言～由于平台限制字数，如果想帮忙debug代码的同学可以到 https://paste.org.cn 粘贴题干描述及代码，生成链接拷贝到本平台发布，这样也方便代码复制和格式调整。希望能帮助到大家！——曹彧 (edited)

Login as host - Present mode
Acceptable Use - Slido Privacy
Cookie Settings
© 2012–2024 slido - 58.44.1
slido
Ask

Polls Result

16 Participants,
32 valid results

感谢同学们的热情互动！



在计算概论中，你感觉最薄弱或者最想听的是哪些部分？（比如“字符串”，“列表”，“循环”，“输入输出”，“进制转换”，“题目分析”等）（一空一词，更多内容可以到 Q&A session 留言！）

16

for循环。题目分析 列表

综合型题目 列表 字符串 循环语句

函数 题目分析 输入输出

综合性题目 循环 进制转换

其实大部分都不太会ww

Table of contents 目录

01

Leading Part

为什么学计概?
怎么才能学好计概?

02

View Python Differently

Python的命令行运行、**断点调试**
Environment Diagram

03

Review Session

基本数据类型、基本程序结构
字符串、列表、元组

04

Practical Exercise

实战分析

01

Leading Part

为什么学计算概论？怎么才能学好计算概论？

Why Computation?

CS61A in Berkeley
2000+ participants!

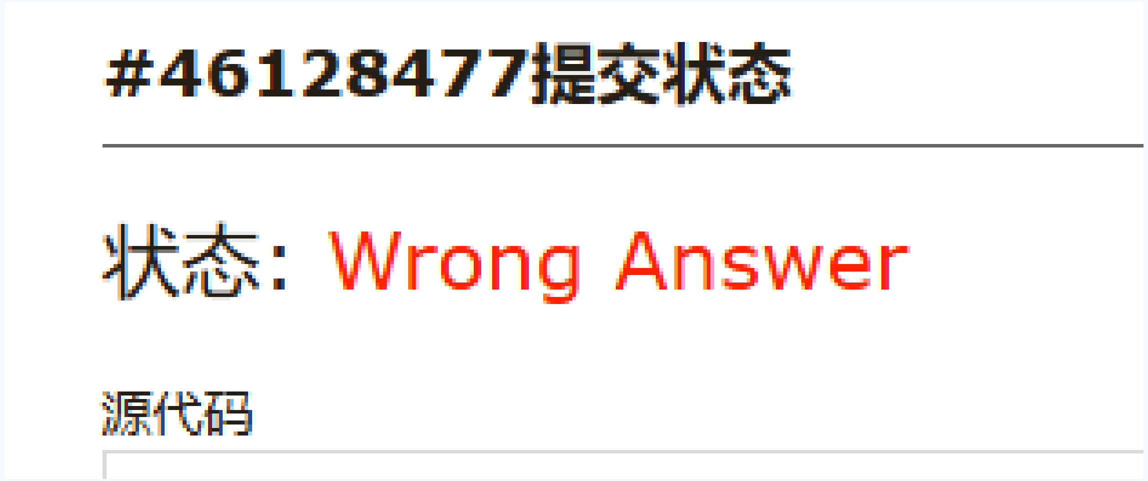
Because… Programming is Cool!

- On song editing
- On dealing with images
- On Microsoft word, excel, etc.

Python is playing its role on everything!

On Dealing With Images

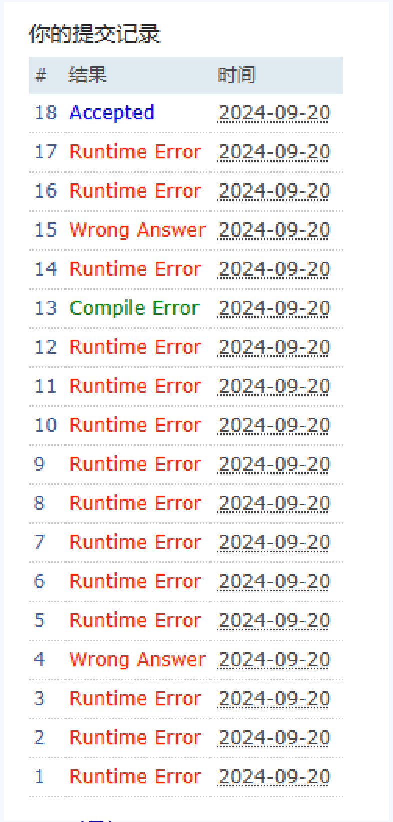
But… Why are we losing passion?



#46128477提交状态

状态: Wrong Answer

源代码



你的提交记录

#	结果	时间
18	Accepted	2024-09-20
17	Runtime Error	2024-09-20
16	Runtime Error	2024-09-20
15	Wrong Answer	2024-09-20
14	Runtime Error	2024-09-20
13	Compile Error	2024-09-20
12	Runtime Error	2024-09-20
11	Runtime Error	2024-09-20
10	Runtime Error	2024-09-20
9	Runtime Error	2024-09-20
8	Runtime Error	2024-09-20
7	Runtime Error	2024-09-20
6	Runtime Error	2024-09-20
5	Runtime Error	2024-09-20
4	Wrong Answer	2024-09-20
3	Runtime Error	2024-09-20
2	Runtime Error	2024-09-20
1	Runtime Error	2024-09-20

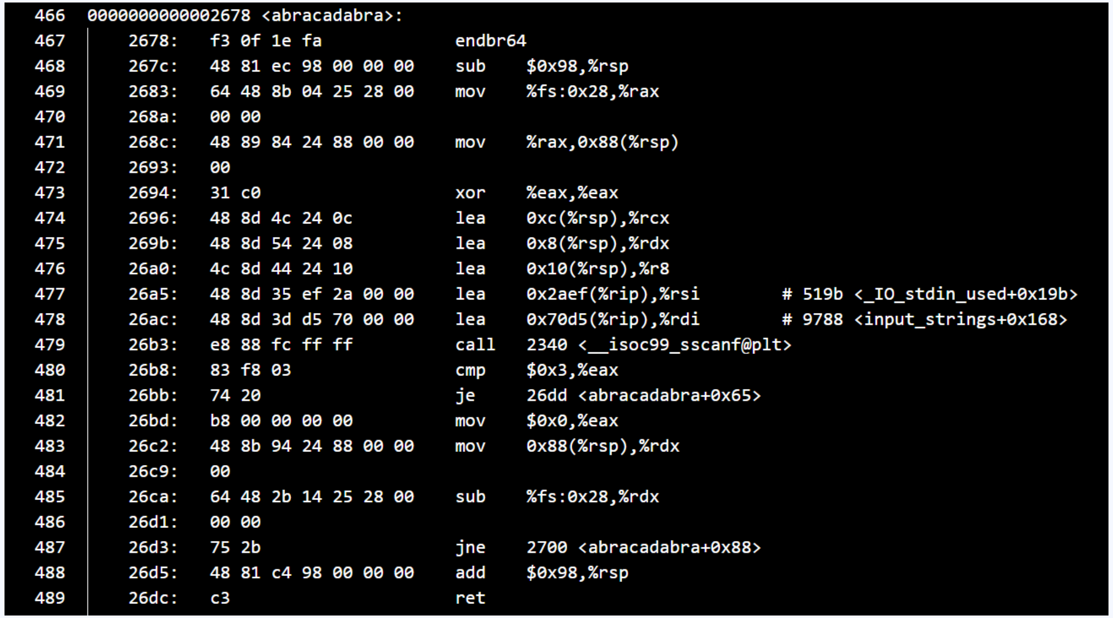
The Key Word in Computer Science

抽象

In English: Abstraction(a noun, not an adjective!)

Without Abstraction?

The Code Would Be Like This!



```
0000000000002678 <abracadabra>:
    2678:   f3 0f 1e fa             endbr64
    267c:   48 81 ec 98 00 00 00    sub    $0x98,%rsp
    2683:   64 48 8b 04 25 28 00    mov    %fs:0x28,%rax
    268a:   00 00
    268c:   48 89 84 24 88 00 00    mov    %rax,0x88(%rsp)
    2693:   00
    2694:   31 c0                   xor    %eax,%eax
    2696:   48 8d 4c 24 0c          lea    0xc(%rsp),%rcx
    269b:   48 8d 54 24 08          lea    0x8(%rsp),%rdx
    26a0:   4c 8d 44 24 10          lea    0x10(%rsp),%r8
    26a5:   48 8d 35 ef 2a 00 00    lea    0x2aef(%rip),%rsi        # 519b <_IO_stdin_used+0x19b>
    26ac:   48 8d 3d d5 70 00 00    lea    0x70d5(%rip),%rdi        # 9788 <input_strings+0x168>
    26b3:   e8 88 fc ff ff          call   2340 <__isoc99_sscanf@plt>
    26b8:   83 f8 03                cmp    $0x3,%eax
    26bb:   74 20                   je     26dd <abracadabra+0x65>
    26bd:   b8 00 00 00 00          mov    $0x0,%eax
    26c2:   48 8b 94 24 88 00 00    mov    0x88(%rsp),%rdx
    26c9:   00
    26ca:   64 48 2b 14 25 28 00    sub    %fs:0x28,%rdx
    26d1:   00 00
    26d3:   75 2b                   jne    2700 <abracadabra+0x88>
    26d5:   48 81 c4 98 00 00 00    add    $0x98,%rsp
    26dc:   c3                      ret
```

Data are all **Bits…** We need to “abstract” them

So Python is a **helper**, not an **enemy**!

“A language isn't something you learn so much as something you join.”

—CS61A Textbook

Some Suggestions!

1. Enjoy Python!
2. 要“动起来”，不要仅“格”幻灯片和讲义
3. 尽量少用AI等辅助工具，而是养成独立思考的习惯
4. 要有一定量的**题目练习**，做题很关键
5. 下学期的课程进度会逐渐加快，函数、递归的部分可能会有挑战性，因此**Try your best to follow up!** 遇到不会的问题及时向老师、TA、其他同学求助

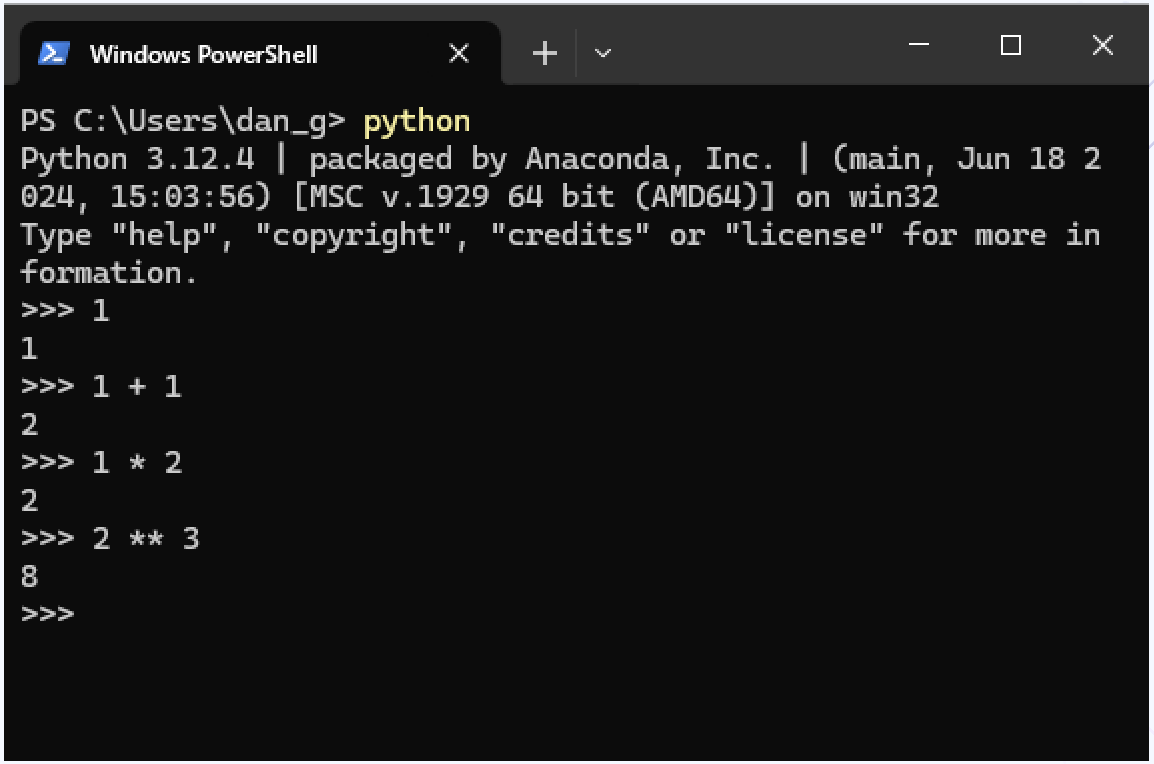
02

View Python

Python的命令行运行、断点调试、Environment Diagram

Python interactive mode（命令行）

(Demo here)



```
PS C:\Users\dan_g> python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2
024, 15:03:56) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more in
formation.
>>> 1
1
>>> 1 + 1
2
>>> 1 * 2
2
>>> 2 ** 3
8
>>>
```

Expressions（表达式）

表达式通常描述一段计算过程，并表示一个数值。

在Python中，所有表达式都可以转化为函数调用 (call expression) 的模式。

(Demo!)

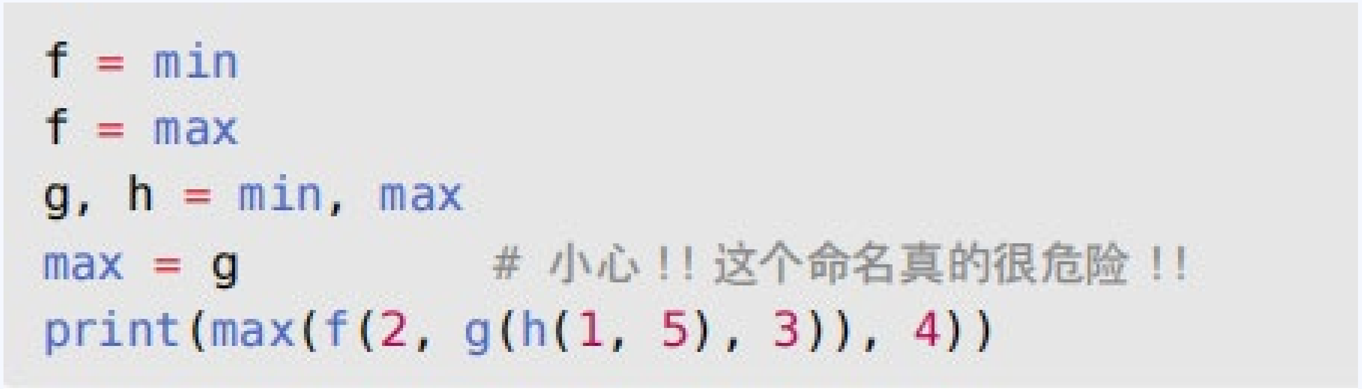
命名和赋值

在Python中，我们可以自己定义变量，并且给变量命名、做赋值操作。

命名是有规范的！好的命名可以显著提高编程效率，降低Bug出现的几率。

Challenge Task! （论命名不规范的下场）

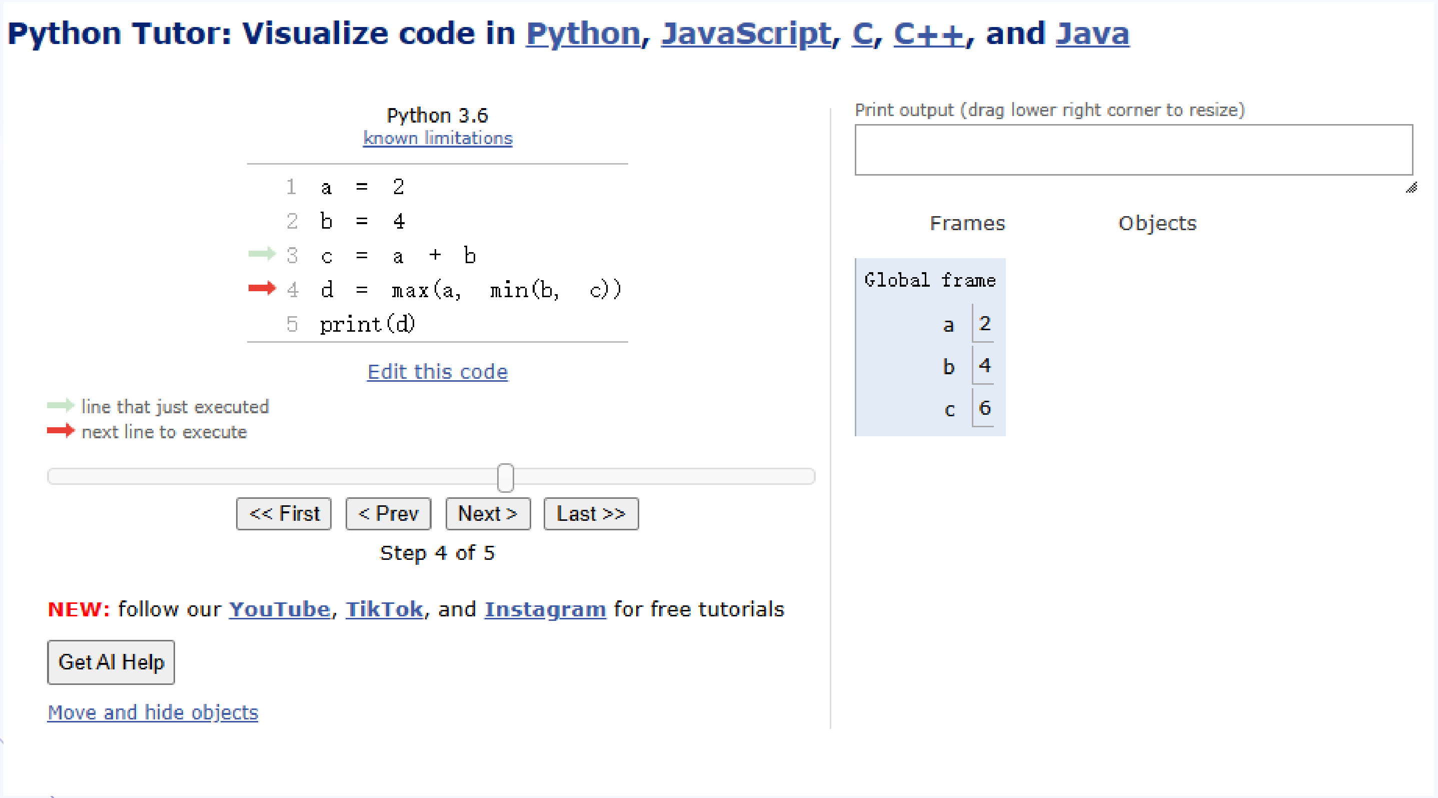
请问下面的代码输出什么？



```
f = min
f = max
g, h = min, max
max = g             # 小心!! 这个命名真的很危险!!
print(max(f(2, g(h(1, 5), 3)), 4))
```

（答案后面会揭晓）

Pythontutor.com



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java

Python 3.6
known limitations

```
a = 2
b = 4
c = a + b
d = max(a, min(b, c))
print(d)
```

Edit this code

Step 4 of 5

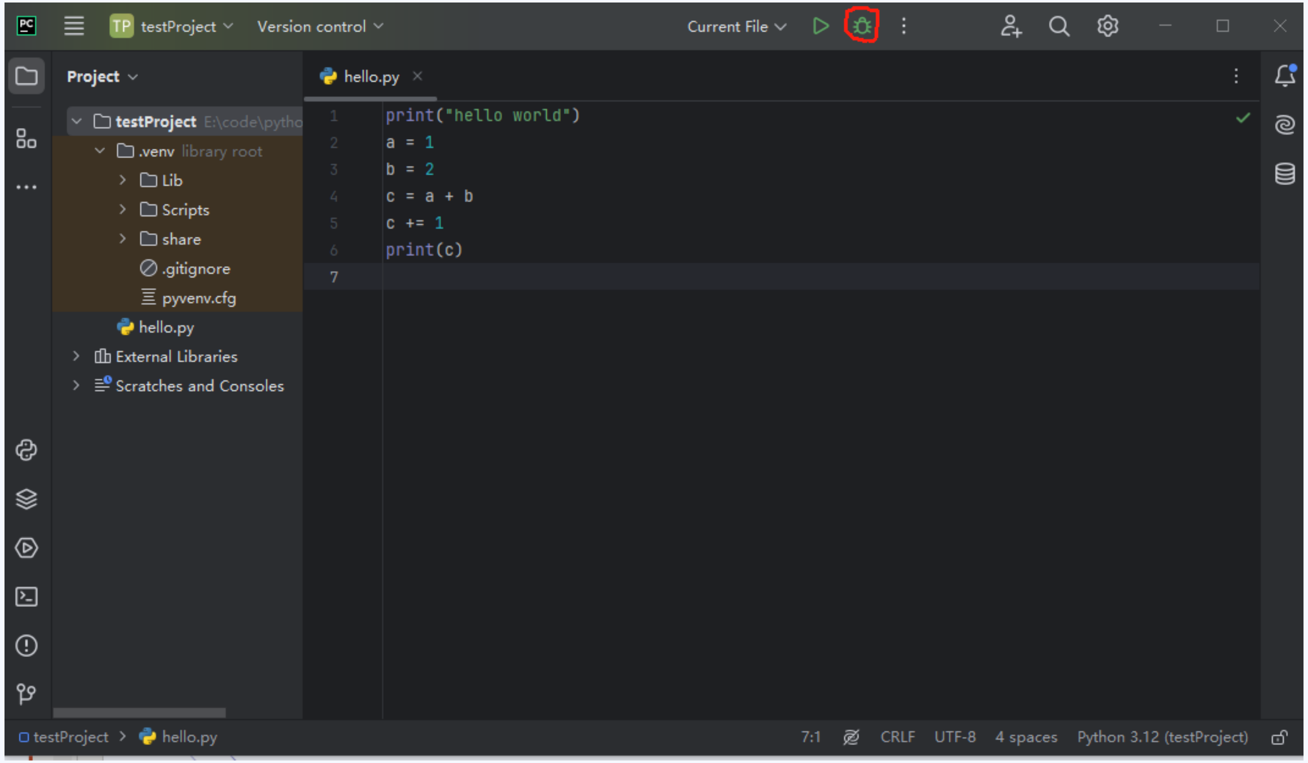
a	2
b	4
c	6

NEW: follow our YouTube, TikTok, and Instagram for free tutorials

Demo1

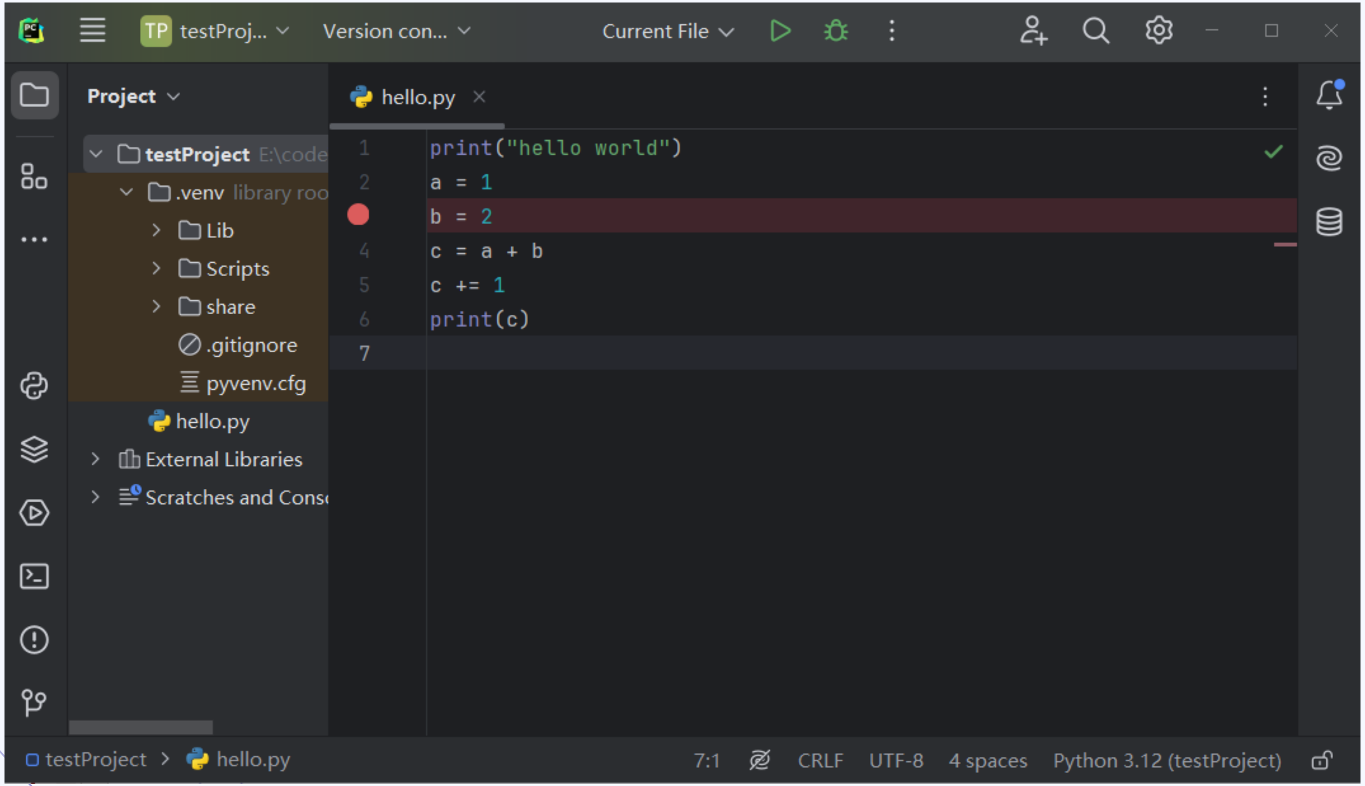
Demo2

Pycharm Debugging Mode



```python
print("hello world")
a = 1
b = 2
c = a + b
c += 1
print(c)
```

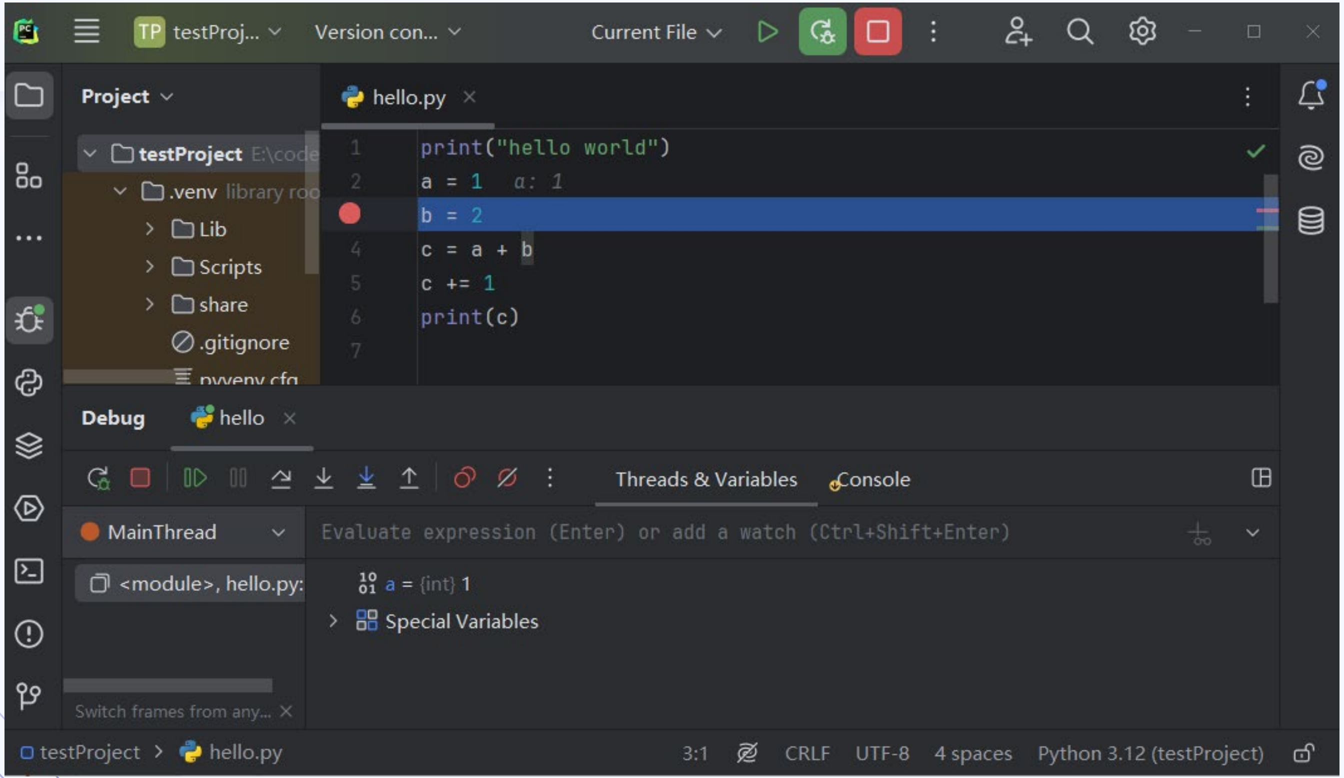
Pycharm Debugging Mode



```
print("hello world")
a = 1
b = 2
c = a + b
c += 1
print(c)
```

断点
Breakpoint

Pycharm Debugging Mode

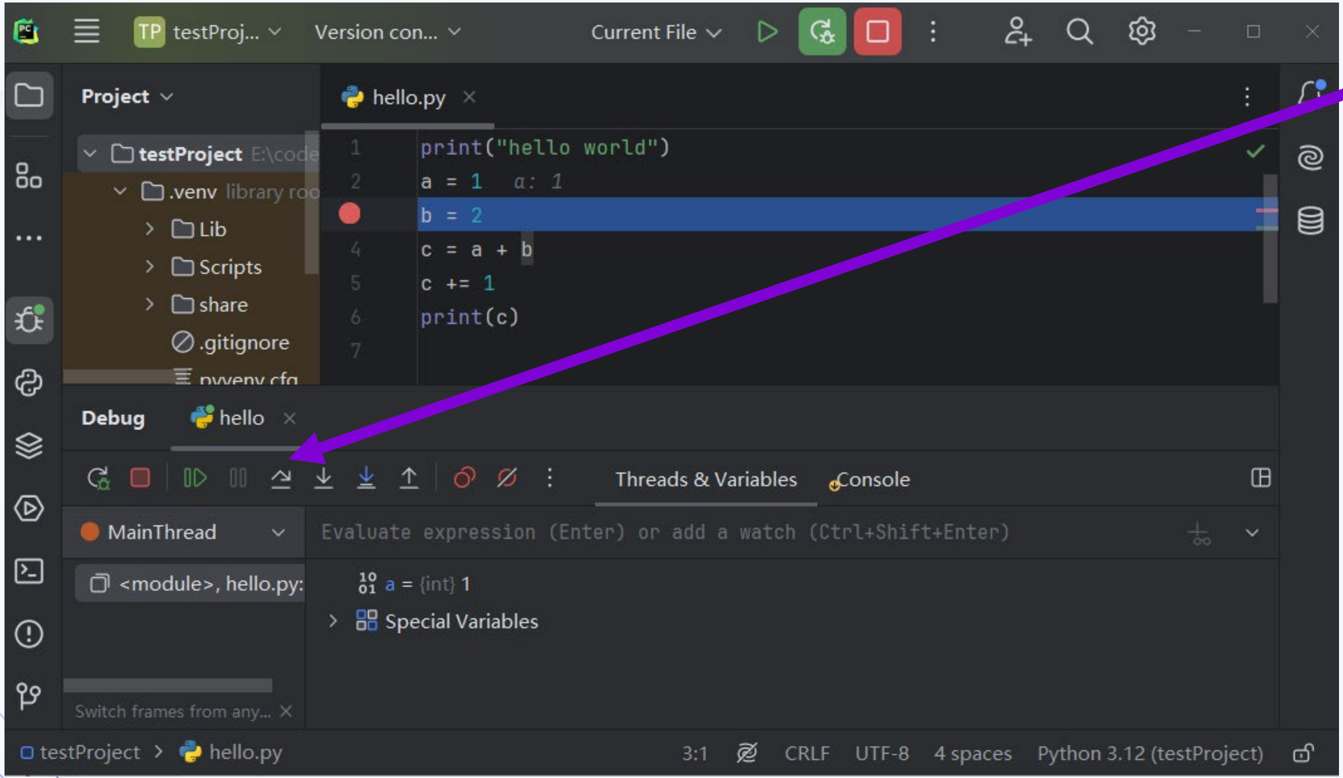


```
print("hello world")
a = 1   a: 1
b = 2
c = a + b
c += 1
print(c)
```

```
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
a = {int} 1
> Special Variables
```

程序开始debug模式后，会自动跳转到第一个断点处暂停

Pycharm Debugging Mode



```
print("hello world")
a = 1   a: 1
b = 2
c = a + b
c += 1
print(c)
```

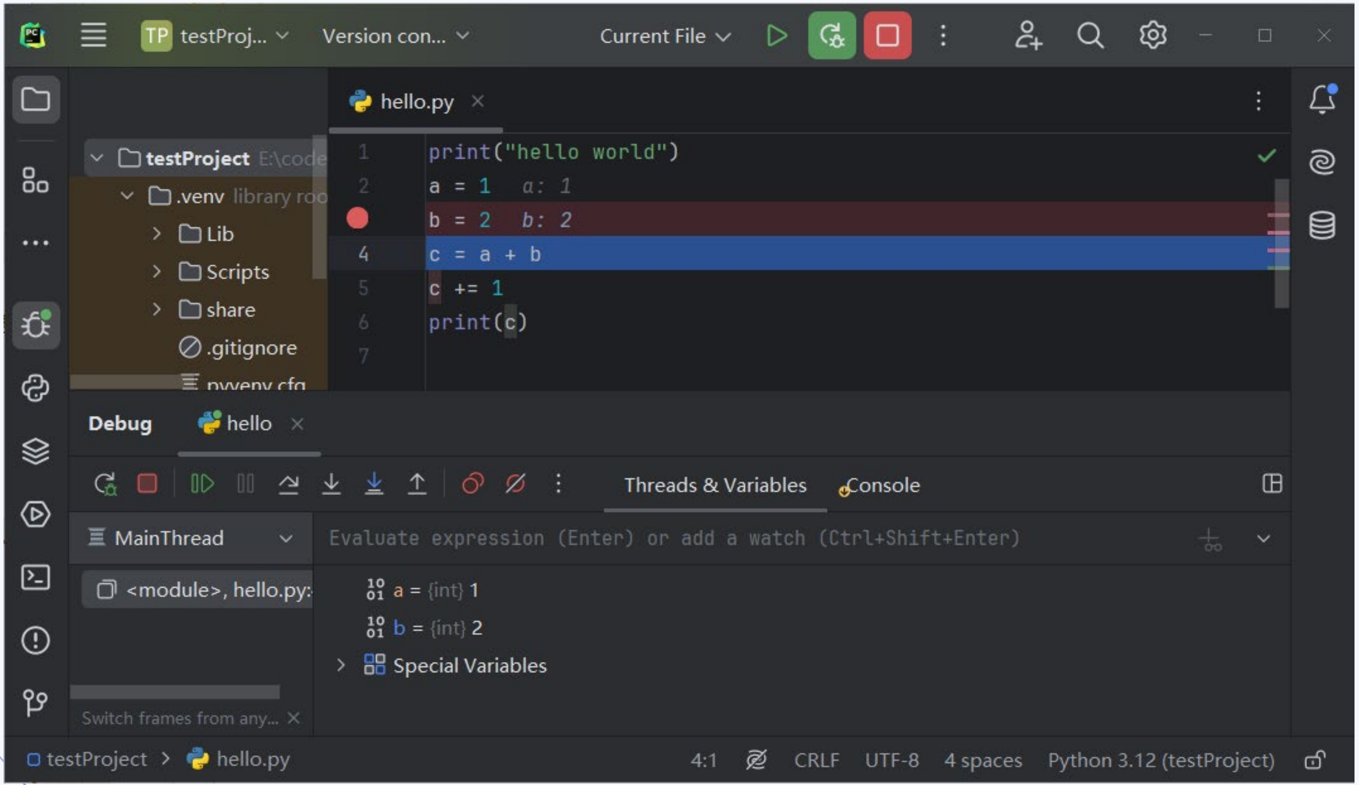
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

a = {int} 1

Special Variables

点这个按钮，会跳转到下一步

Pycharm Debugging Mode

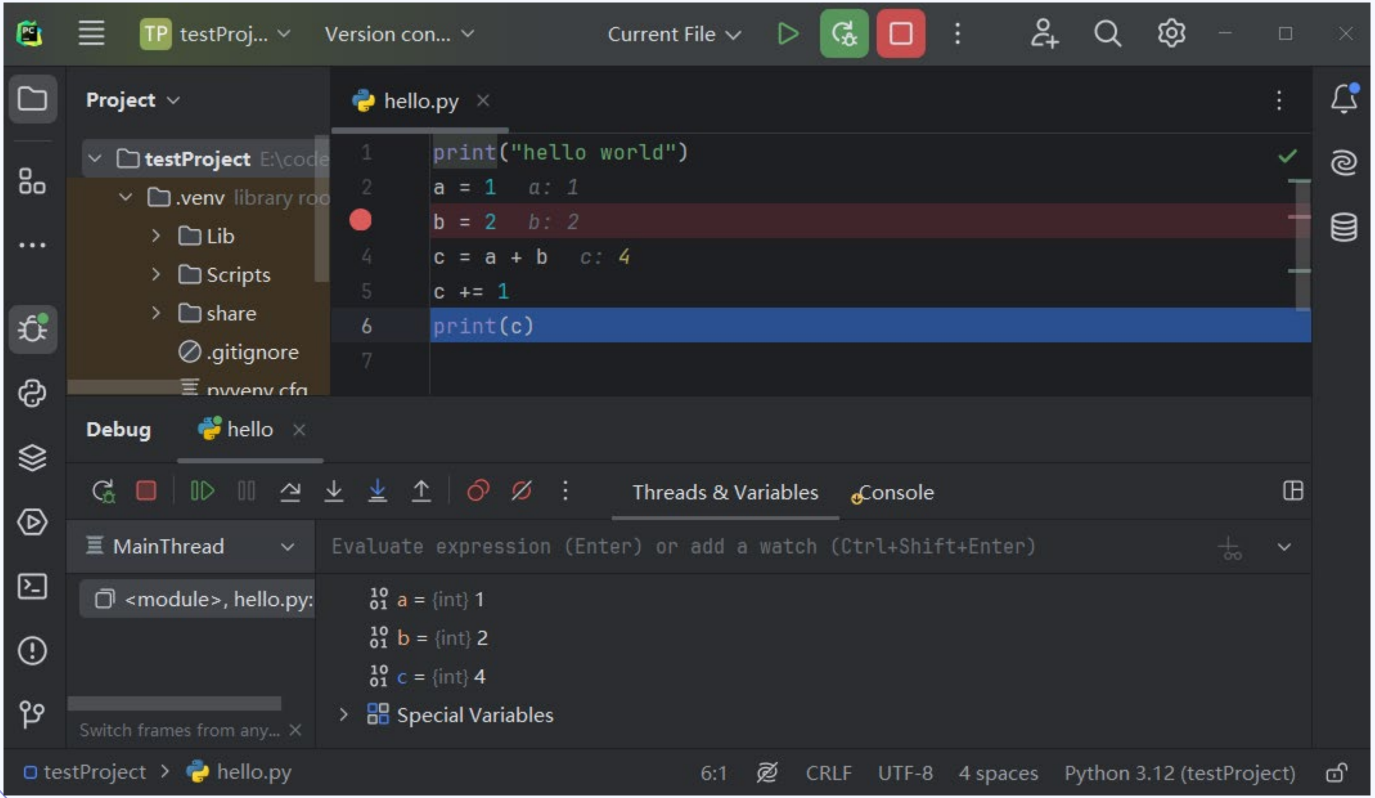


```
print("hello world")
a = 1   a: 1
b = 2   b: 2
c = a + b
c += 1
print(c)
```

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

```
a = {int} 1
b = {int} 2
Special Variables
```

Pycharm Debugging Mode



```
print("hello world")
a = 1   a: 1
b = 2   b: 2
c = a + b   c: 4
c += 1
print(c)
```

03

Review Session

基本数据类型、基本程序结构、字符串、列表、元组

输入和输出

Python的输出函数是print，输入函数是input。

input会读取用户输入的一行，返回类型是**字符串**。

如果想要把该输入字符串转化成其他类型变量，则需要使用**字符串处理**中的一系列函数（后面讲到）

(Demo!)

基本数据类型

int, float, bool

基本程序结构

顺序，分支，循环

序列 Sequence

序列是Python中的一种很重要的类型，**是一些值的有序排列**

常见的种类：list, tuple, str, range, ...

尽管序列有很多种，但他们有很多共同的属性。

len: 序列的长度

[ ]: 选择元素、切片

in: 判断是否在其中

for: for item in sequence循环

序列 Sequence

序列是Python中的一种很重要的类型，**是一些值的有序排列**

常见的种类：list, tuple, str, range, ...

尽管序列有很多种，但他们有很多共同的属性。

对于装有int和float的序列: sum, max, min

对于list, tuple, str: +, * 序列变长

序列的map操作

map(func, sequence)

相当于用函数func作用于sequence的每一个元素

返回的是一个叫做iterator的东西…（不过iterator超纲了）只需要知道它可以转换成一个序列就好

```
for index in range(len(sequence)):
        sequence[index] = func(sequence[index])
```

Exercise: 解释list(map(int, input().split()))

字符串的特殊操作

str.find(substr): 找到substr的第一次出现位置

str.split(t=‘ ’): 以t为分割符（默认为空格）分割字符串，返回列表

str.upper(), str.lower(), str.isupper(), str.islower()

字符串的格式化

1. 用%
2. 用str.format
3. 用f“str”

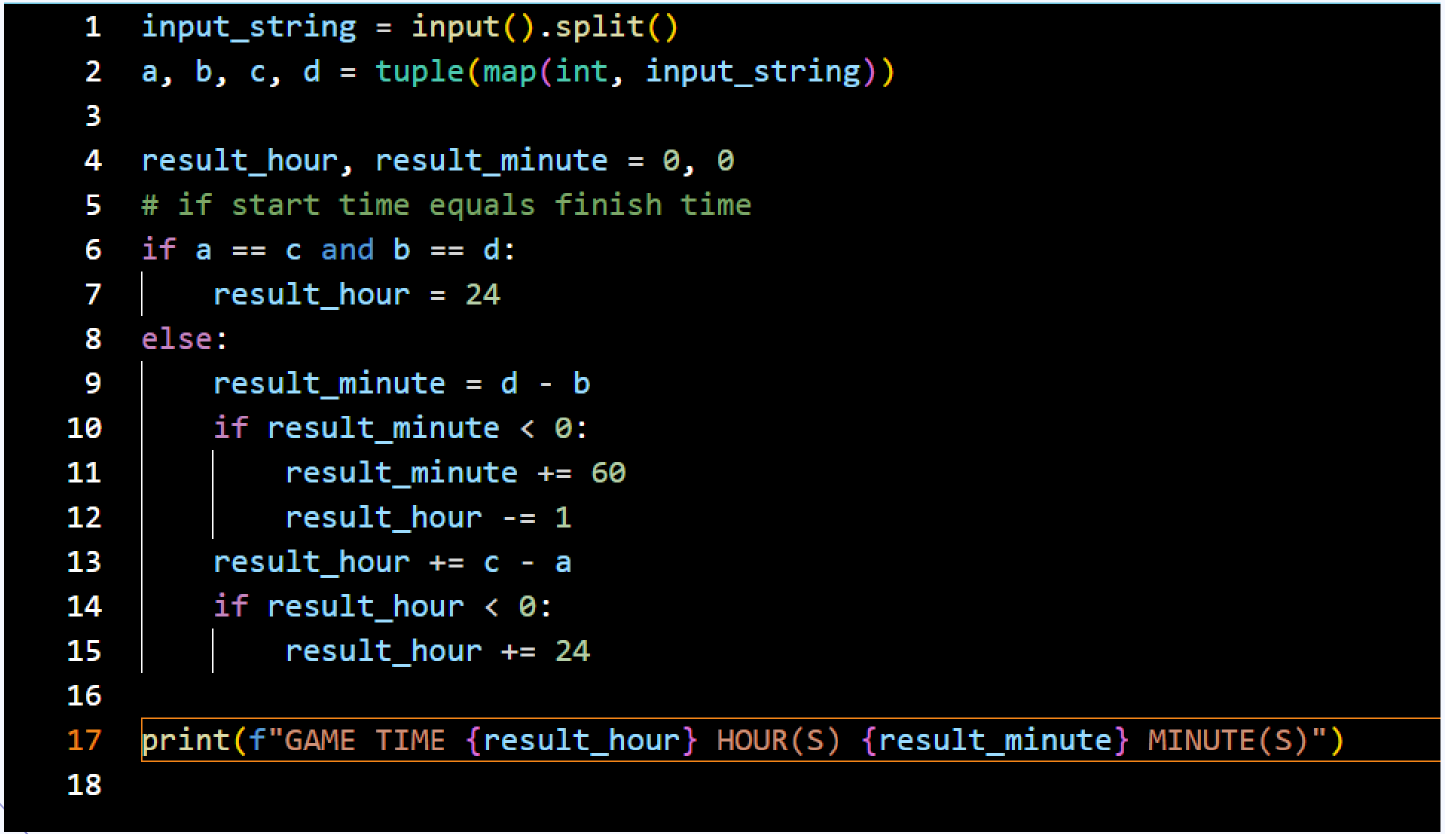
Demo！

04

Exercises

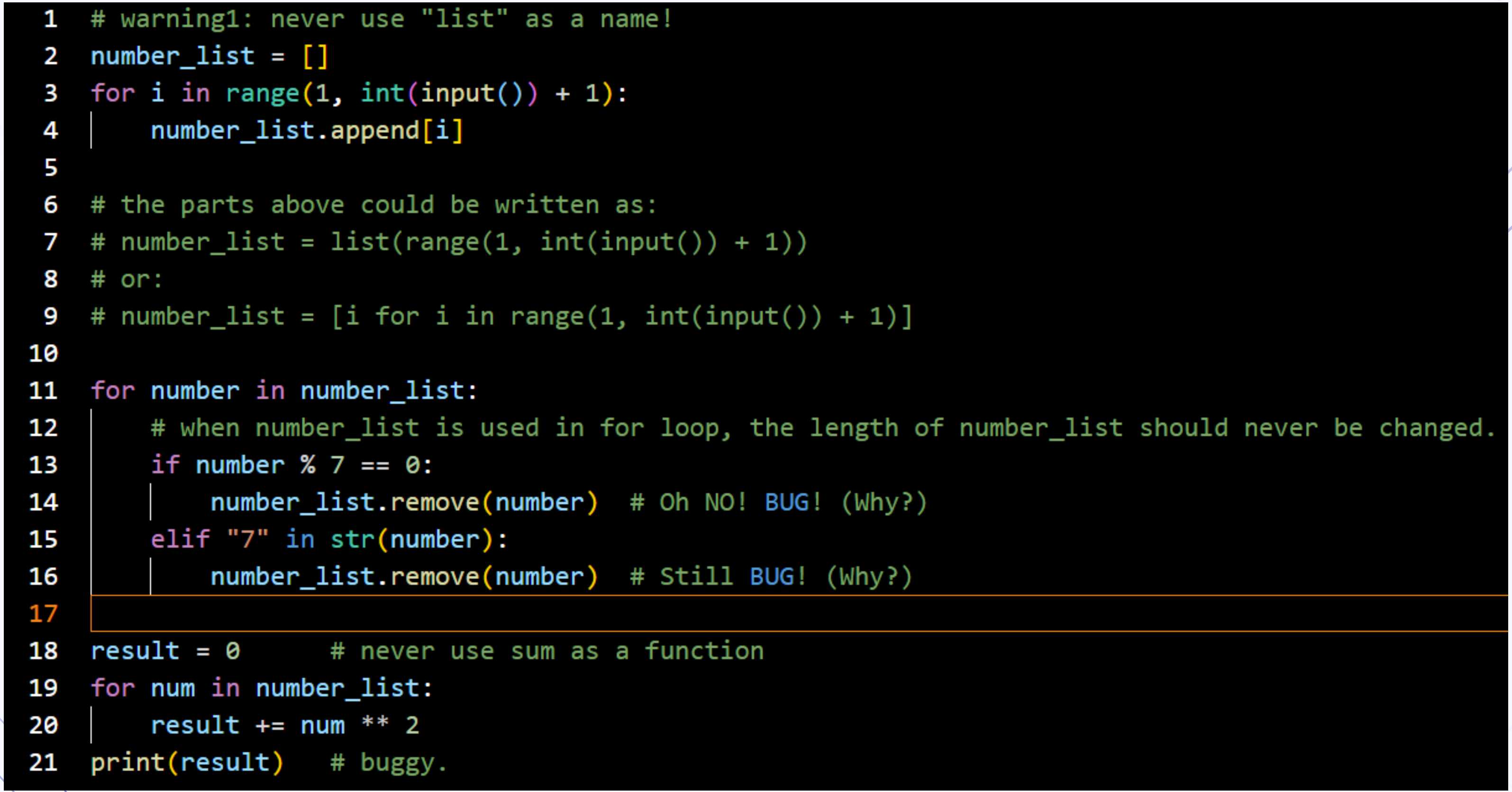
题目见讲义内容

游戏时间



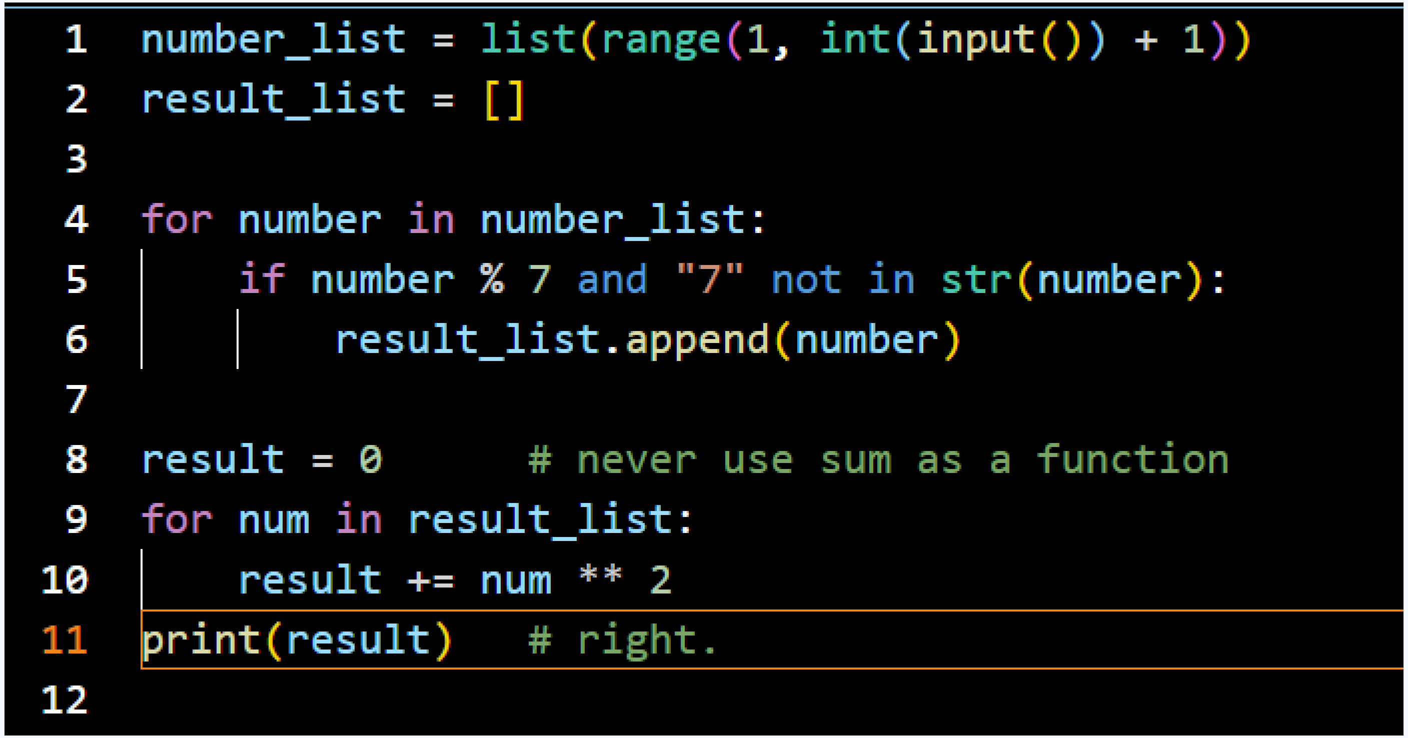
```python
input_string = input().split()
a, b, c, d = tuple(map(int, input_string))

result_hour, result_minute = 0, 0
# if start time equals finish time
if a == c and b == d:
    result_hour = 24
else:
    result_minute = d - b
    if result_minute < 0:
        result_minute += 60
        result_hour -= 1
    result_hour += c - a
    if result_hour < 0:
        result_hour += 24

print(f"GAME TIME {result_hour} HOUR(S) {result_minute} MINUTE(S)")

```

与7无关的数



```python
# warning1: never use "list" as a name!
number_list = []
for i in range(1, int(input()) + 1):
    number_list.append[i]

# the parts above could be written as:
# number_list = list(range(1, int(input()) + 1))
# or:
# number_list = [i for i in range(1, int(input()) + 1)]

for number in number_list:
    # when number_list is used in for loop, the length of number_list should never be changed.
    if number % 7 == 0:
        number_list.remove(number)  # Oh NO! BUG! (Why?)
    elif "7" in str(number):
        number_list.remove(number)  # Still BUG! (Why?)

result = 0      # never use sum as a function
for num in number_list:
    result += num ** 2
print(result)   # buggy.
```

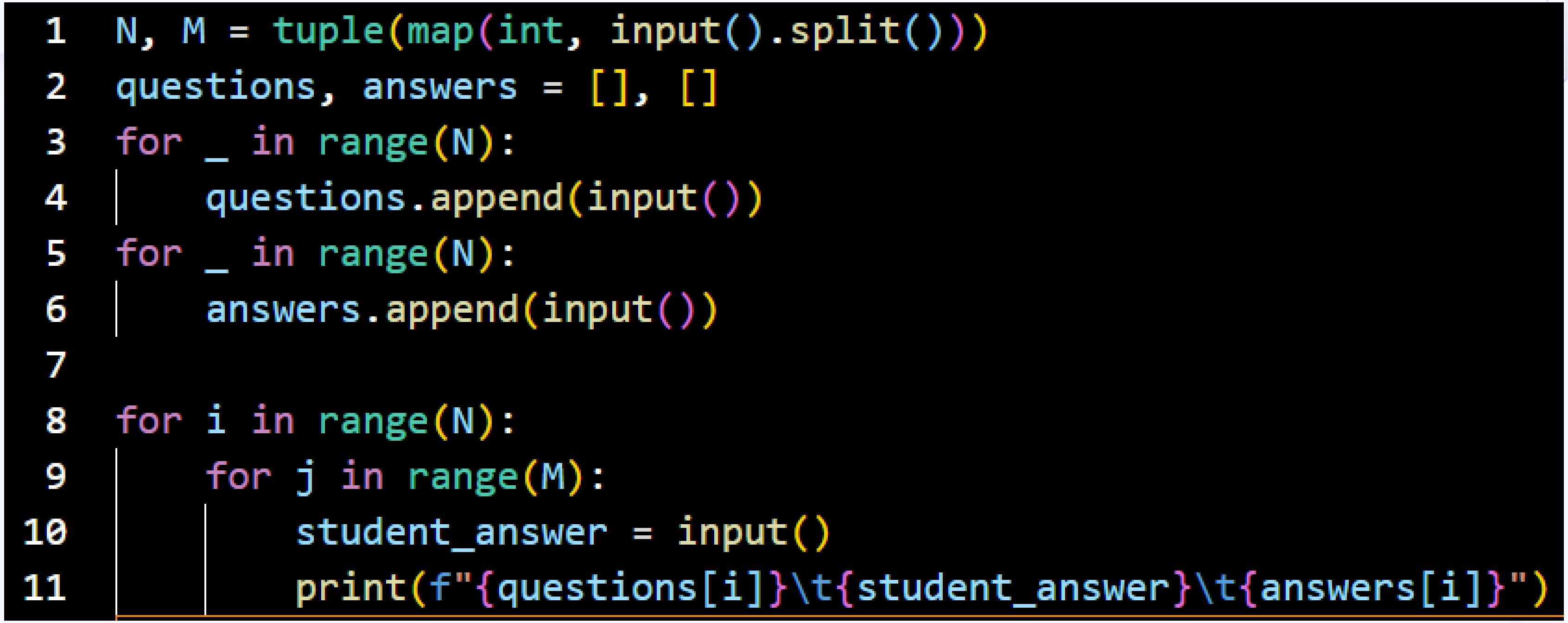
与7无关的数


```python
number_list = list(range(1, int(input()) + 1))
result_list = []

for number in number_list:
    if number % 7 and "7" not in str(number):
        result_list.append(number)

result = 0       # never use sum as a function
for num in result_list:
    result += num ** 2
print(result)   # right.

```

与7无关的数（一行解决！）

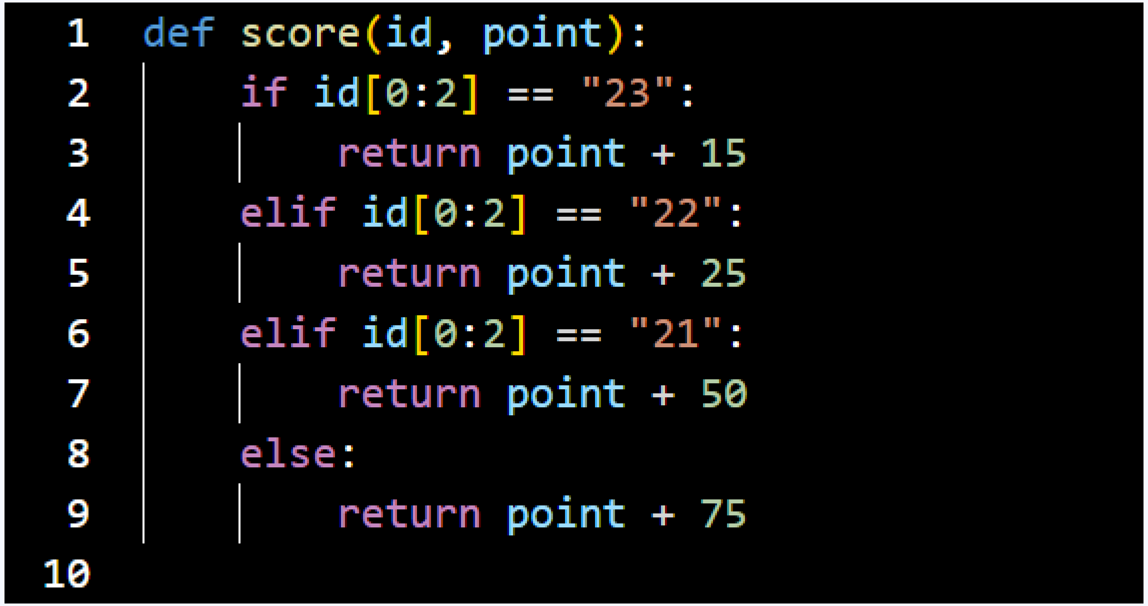
```
print(sum([i ** 2 for i in range(1,
    int(input() + 1)) if i % 7 and "7" not
    in str(i)]))
```

处理答题数据集



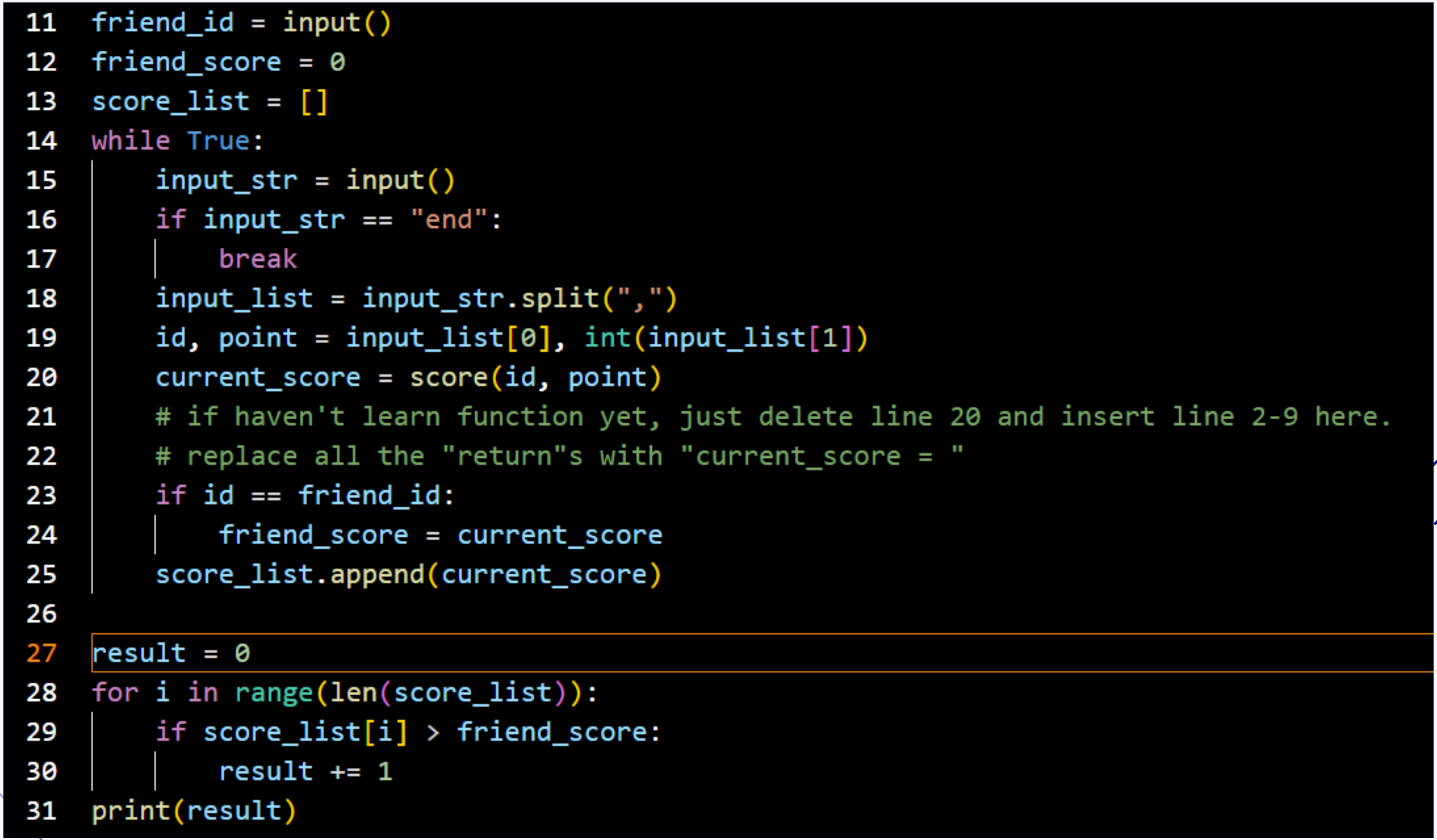
```python
N, M = tuple(map(int, input().split()))
questions, answers = [], []
for _ in range(N):
    questions.append(input())
for _ in range(N):
    answers.append(input())

for i in range(N):
    for j in range(M):
        student_answer = input()
        print(f"{questions[i]}\t{student_answer}\t{answers[i]}")
```

选课意愿点



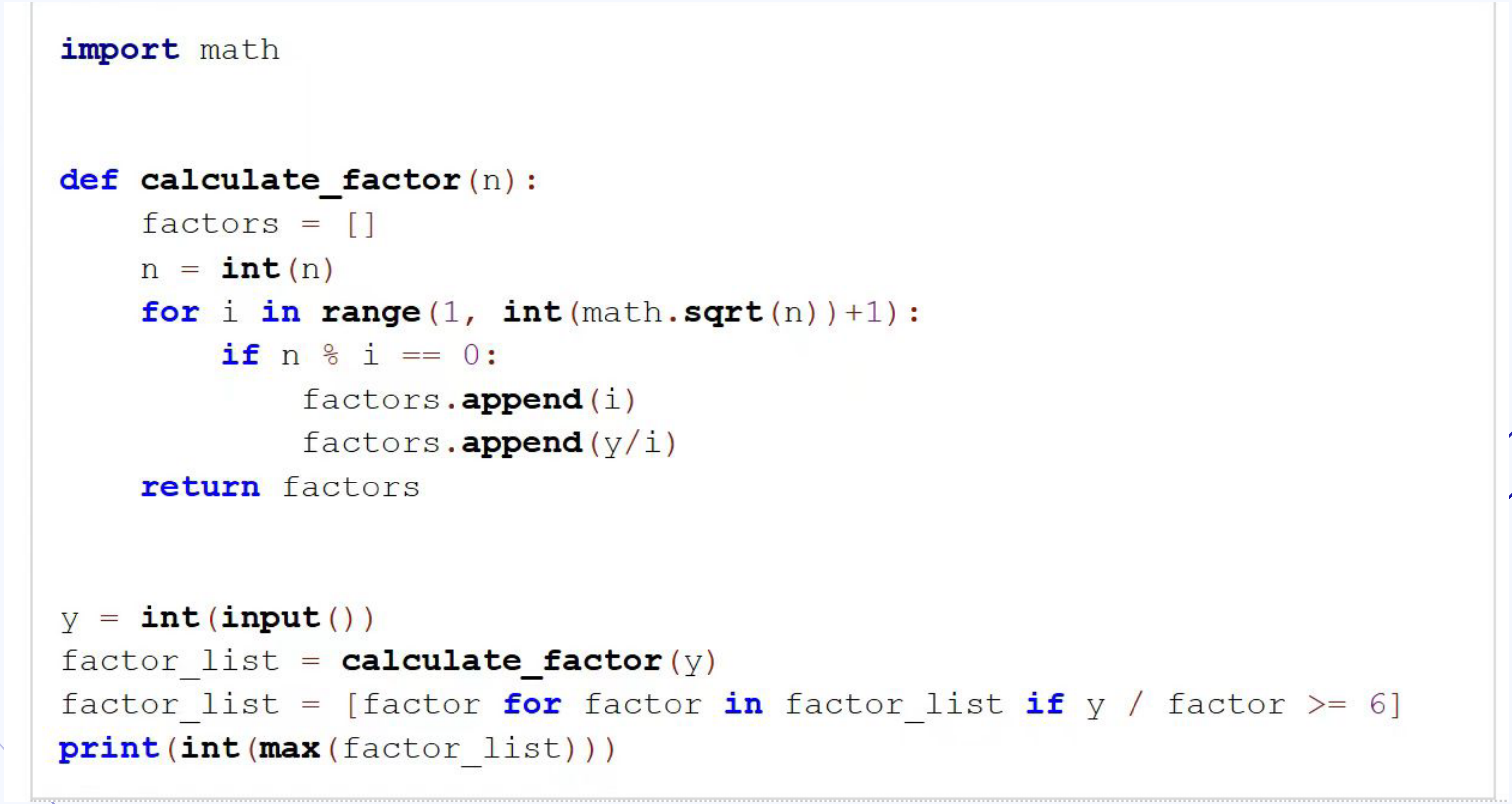
```python
def score(id, point):
    if id[0:2] == "23":
        return point + 15
    elif id[0:2] == "22":
        return point + 25
    elif id[0:2] == "21":
        return point + 50
    else:
        return point + 75

```

选课意愿点



```
11  friend_id = input()
12  friend_score = 0
13  score_list = []
14  while True:
15      input_str = input()
16      if input_str == "end":
17          break
18      input_list = input_str.split(",")
19      id, point = input_list[0], int(input_list[1])
20      current_score = score(id, point)
21      # if haven't learn function yet, just delete line 20 and insert line 2-9 here.
22      # replace all the "return"s with "current_score = "
23      if id == friend_id:
24          friend_score = current_score
25      score_list.append(current_score)
26
27  result = 0
28  for i in range(len(score_list)):
29      if score_list[i] > friend_score:
30          result += 1
31  print(result)
```

数字密码



```
import math


def calculate_factor(n):
    factors = []
    n = int(n)
    for i in range(1, int(math.sqrt(n))+1):
        if n % i == 0:
            factors.append(i)
            factors.append(y/i)
    return factors


y = int(input())
factor_list = calculate_factor(y)
factor_list = [factor for factor in factor_list if y / factor >= 6]
print(int(max(factor_list)))
```

Thanks!

Do you have any questions?

calvincao@stu.pku.edu.cn
+86 186 8665 9012
calvinxiaocao.github.io