```
pi = 3.14
r = 5
c = 10
h = 8
shape = 'triangle'

if shape == 'circle':
  Area = pi * (r ** 2)
  print(Area)

elif shape == 'sq':
  Area = c ** 2
  print(Area)

elif shape == 'triangle':
  Area = 0.5 * c * h
  print(Area)

else:
  Area = 2 * pi * r * h + 2 * pi * r ** 2
  print(Area)
```

```
40.0
```

```
gr = 97
if gr > 96:
  print("A+")
elif gr > 93:
  print("A")
elif gr > 90:
  print("A-")
elif gr > 88:
  print("B+")
```

```
A+
```

```
amount = float(input('Enter investment amount (($)): '))
customer_type = input('Enter type of customer (n-> new, e-> existing): ')

if amount <= 1000 and customer_type == 'e':
  ir = 0.03
  ret = (1 + ir) * amount
elif amount <= 10000 and customer_type == 'e':
  ir = 0.0325
  ret = (1 + ir) * amount
elif amount > 10000 and customer_type == 'e':
  ir = 0.035
  ret = (1 + ir) * amount
else:
  ir = 0.04
  ret = (1 + ir) * amount

print(f'Annual return after 1 year: ${ret:.2f}')
```

```
Enter investment amount (($)): 1000
Enter type of customer (n-> new, e-> existing): n
Annual return after 1 year: $1040.00
```

```
play = input('Would you like to play? (): ')
while play == 'Yes':
  p1 = input('Enter P1: Rock, Paper, Scissors (): ')
  p2 = input('Enter P2: Rock, Paper, Scissors (): ')
  if p1 == p2:
    print("Draw")
  elif (p1 == "Rock" and p2 == "Scissors") or \
     (p1 == "Scissors" and p2 == "Paper") or \
     (p1 == "Paper" and p2 == "Rock"):
    print("P1 Wins!")
  elif (p1 == "Rock" and p2 == "Paper") or \
     (p1 == "Scissors" and p2 == "Rock") or \
     (p1 == "Paper" and p2 == "Scissors"):
    print("P2 Wins!")

  play = input('Would you like to play? () ')
```

```
n1 = float(input("number 1: "))
n2 = float(input("number 2: "))
symbol = input("operation: ")

if symbol == "+":
  ans = n1 + n2
  print(ans)
elif symbol == "-":
  ans = n1 - n2
  print(ans)
elif symbol == "*":
  ans = n1 * n2
  print(ans)
elif symbol == "/":
  ans = n1 / n2
  print(ans)
elif symbol == "**":
  ans = n1 ** n2
  print(ans)
```

```
number 1: 2
number 2: 4
operation: **
16.0
```

```
flat_rate = 5
hourly_rate = 2.5

for h in range(1, 9, 1):
  charge = flat_rate + hourly_rate * h
  if charge < 10:
    charge = 10
  elif charge > 20:
    charge = 20
  print (h, charge)
```

```
1 10
2 10.0
3 12.5
4 15.0
5 17.5
6 20.0
7 20
8 20
```

```
ticketPrice = 10
fixedCost = 200
fixedAttendees = 20

for ad in range(0, 201, 25):
  additionalAttendees = round(2 * ad ** 0.5)
  profit = (fixedAttendees + additionalAttendees) * ticketPrice - ad - fixedCost
  print(ad, profit)
```

```
0 0
25 75
50 90
75 95
100 100
125 95
150 90
175 85
200 80
```

```
from ast import NameConstant
from ast import NameConstant
```

```
months = {}

names = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
numbers = [1,2,3,4,5,6,7,8,9,10,11,12]

for n in range(1,13):
  months[n] = names[n-1]

print(months)
```

```
{1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May', 6: 'Jun', 7: 'Jul', 8: 'Aug', 9: 'Sep', 10: 'Oct', 11: 'Nov', 12: 'Dec'}
```

```
check = input('Enter yes or no: ')
while check == 'Yes':
  print('Hello!')
  check = input('Do you wish to continue: ')
print('Process finished')
```

```
Enter yes or no: Yes
Hello!
Do you wish to continue: Yes
Hello!
Do you wish to continue: No
Process finished
```

```
def quickSort(L):
  if len(L) <= 1:
    return L
  else:
    pivot = L[len(L) // 2]

    less = []
    equal = []
    more = []

    for el in L:
      if el < pivot:
        less.append(el)
      elif el == pivot:
        equal.append(el)
      else:
        more.append(el)

  return quickSort(less) + equal + quickSort(more)

quickSort([8,2,5,1,7,9])
```

```
[1, 2, 5, 7, 8, 9]
```

```
def wAvg(L, w):
  try:
    res = []
    for i in range(len(L)):
      res.append(L[i] * w[i])
    return sum(res) / sum(w)
  except ZeroDivisionError:
    print('Divided by zero')
    return []
  except TypeError:
    print('Nun-numerical objects found')
    return []
  except Exception as e:
    print(f'Unknown error: (e)')
    return float('nan')
wAvg([1,2,3], [.2, 0, -.2])
```

```
Divided by zero
[]
```

```
class student:
  def __init__(self, name, number):
    self.__name = name
    self.__number = number
    self.courses = []

  def enroll(self,newCourse):
    if newCourse not in self.__courses:
      self.__courses.append(newCourse)
    else:
      print('You have already enrolled in (newCourse)')

  def get_courses(self):
    return self.courses
```

```
total_cost = float(input("Enter the total cost of the house: "))
portion_down_payment = 0.25
current_savings = 0.0
r = 0.04
annual_salary = float(input("Enter your annual salary: "))
portion_saved = float(input("Enter the portion of salary to save (as a decimal): "))

down_payment = portion_down_payment * total_cost
monthly_salary = annual_salary / 12
months = 0

while current_savings < down_payment:
   current_savings += current_savings * (r / 12)
   current_savings += portion_saved * monthly_salary
   months += 1

print("Number of months: ", months)
```

```
Enter the total cost of the house: 10000000
Enter your annual salary: 100000
Enter the portion of salary to save: 50000
Number of months:  1
```

```
total_cost = float(input("Enter the total cost of the house: "))
portion_down_payment = 0.25
current_savings = 0.0
r = 0.04
annual_salary = float(input("Enter your annual salary: "))
portion_saved = float(input("Enter the portion of salary to save (as a decimal): "))
semi_annual_raise = float(input("Enter the semi annual raise (as a decimal): "))

down_payment = portion_down_payment * total_cost
monthly_salary = annual_salary / 12
months = 0

while current_savings < down_payment:
   current_savings += current_savings * (r / 12)
   current_savings += portion_saved * monthly_salary
   months += 1

   if months % 6 == 0:
     annual_salary += annual_salary * semi_annual_raise
     monthly_salary = annual_salary / 12

print("Number of months: ", months)
```

```
Enter the total cost of the house: 120000
Enter your annual salary: 50000
Enter the portion of salary to save (as a decimal): 0.1
Enter the semi annual raise (as a decimal): 0.03
Number of months:  58
```

```
def range_check_print(num, low, high):
    if low <= num <= high:
      print(f"{num} is in the range between {low} and {high}.")
    else:
      print(f"{num} is NOT in the range between {low} and {high}.")

range_check_print(10,5,15)
```

```
10 is in the range between 5 and 15.
```