Programming Fundamental

A Hitchhiker Guide to Coding with Python

Lesson 3: Logical Conditions

Akarate Singta

Department of Mathematics
Faculty of Science and Technology, RMUTT

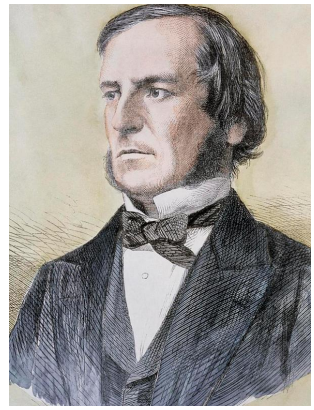
Lesson Outline

Boolean Data Type

A largely self-taught English mathematician, philosopher, and logician, most of whose short career was spent as the first professor of mathematics at Queen's College, Cork in Ireland.

He worked in the fields of differential equations and algebraic logic, and is best known as the author of The Laws of Thought (1854) which contains Boolean algebra.

Boolean logic is credited with laying the foundations for the Information Age



George Boole

(2 Nov 1815 - 8 Dec 1864)

Class	Members	Data Type
`bool`	`True`, `False`	Boolean

Boolean Operators

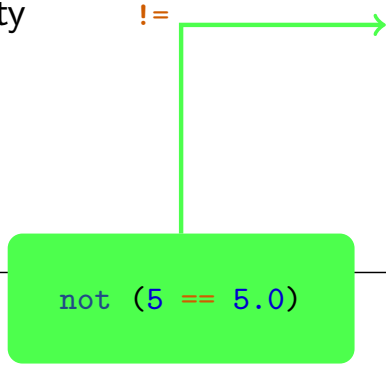
Operation	Operator	Example	Result
Conjunction	`and`	`True and True`	`True`
		`True and False`	`False`
		`False and True`	`False`
		`False and False`	`False`
Disjunction	`or`	`True or True`	`True`
		`True or False`	`True`
		`False and True`	`True`
		`False and False`	`False`
Negation	`not`	`not True`	`False`
		`not False`	`True`

Comparison Operators

Operation	Operator	Example	Result
Equality	==	`5 == 5.0`	`True`
		`-1.255 == -1.25`	`False`
		`'Hello' == 'Hello '`	`False`
		`'A' == 'a'`	`False`
Inequality	!=	`5 != 5.0`	`False`
		`-1.255 != -1.25`	`True`
		`'Hello' != 'Hello '`	`True`
		`'A' != 'a'`	`True`

Comparison Operators

Operation	Operator	Example	Result
Equality	`==`	`5 == 5.0`	`True`
		`-1.255 == -1.25`	`False`
		`'Hello' == 'Hello '`	`False`
		`'A' == 'a'`	`False`
Inequality	`!=`	`5 != 5.0`	`False`
		`-1.255 != -1.25`	`True`
		`'Hello' != 'Hello '`	`True`
		`'A' != 'a'`	`True`



Comparison Operators

Operation	Operator	Example	Result
Strict Ordinalities	<	`5 < 5.14`	`True`
(Less than)		`-3.50 < -4.59`	`False`
		`1 < 1`	`False`
		`-1 < 0 < 1`	`True`
		`'a' < 'b'`	`True`
		`'BNK' < 'BNA'`	`False`

Comparison Operators

Operation	Operator	Example	Result
Strict Ordinalities	<	`5 < 5.14`	`True`
(Less than)		`-3.50 < -4.59`	`False`
		`1 < 1`	`False`
		`-1 < 0 < 1`	`True`
		`'a' < 'b'`	`True`
		`'BNK' < 'BNA'`	`False`

`(-1 < 0) and (0 < 1)`

Comparison Operators

Lexicographical ordering for strings uses the Unicode code point number to order individual characters.

```
>>> ord('a')
97
>>> ord('b')
98
>>> ord('A')
65
>>> ord('B')
66
>>> ord('K')
75
```

```
>>> 'a' < 'b'
True
>>> 'A' < 'a'
True
>>> 'BNK' < 'BNA'
False
```

Comparison Operators

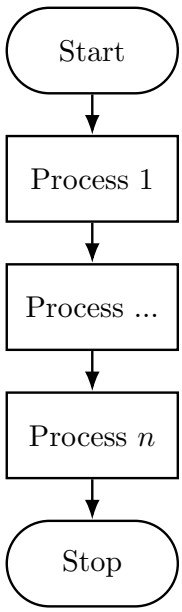
Operation	Operator	Example	Result
Strict Ordinalities	`>`	`5 > 5.14`	`False`
(Greater than)		`-3.50 > -4.59`	`True`
		`1 > 1`	`False`
		`999 > 99 > 9`	`True`
		`'a' > 'b'`	`False`
		`'BNK' > 'BNA'`	`True`

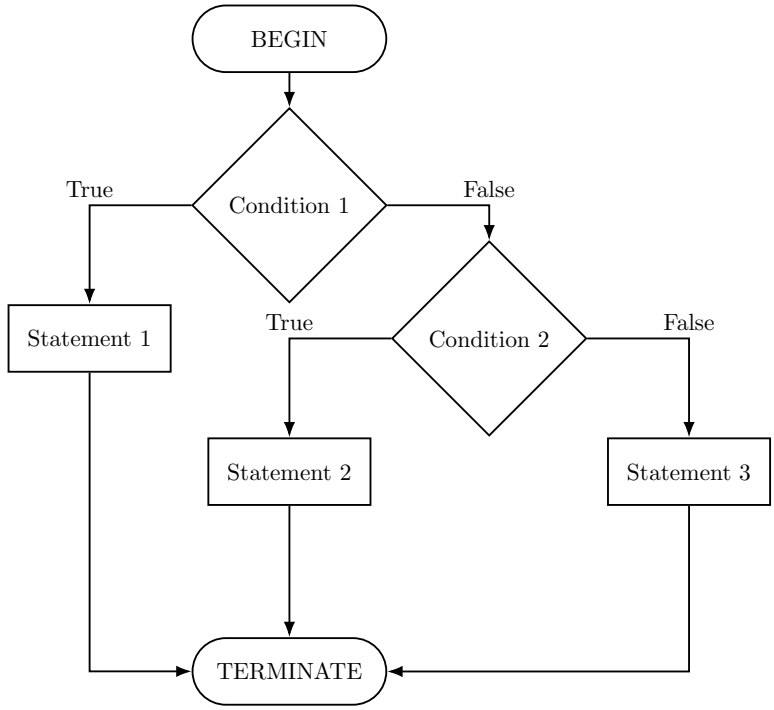
Comparison Operators

Operation	Operator	Example	Result
Ordinalities	`<=`	`5 <= 5.14`	`True`
(Less than or equal to)		`-3.50 <= -4.59`	`False`
		`1 <= 1`	`True`
		`-1 <= 0 <= 1`	`True`
		`'a' <= 'b'`	`True`
		`'BNK' <= 'BNA'`	`False`

Comparison Operators

Operation	Operator	Example	Result
Ordinalities	>=	`5 >= 5.14`	`False`
(Greater than or equal to)		`-3.50 >= -4.59`	`True`
		`1 >= 1`	`True`
		`999 >= 99 >= 9`	`True`
		`'a' >= 'b'`	`False`
		`'BNK' >= 'BNA'`	`True`

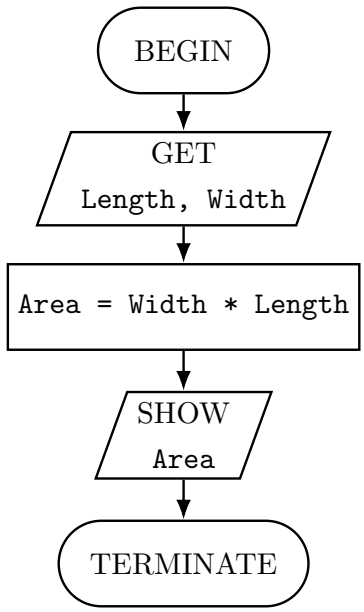

Start
Process 1
Process ...
Process n
Stop


BEGIN
Condition 1
True
False
Statement 1
Condition 2
True
False
Statement 2
Statement 3
TERMINATE

Procedural Programming

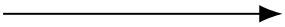
rectangle_calculator.py

```
1  Length = float(input('Enter the length: '))
2  Width = float(input('Enter the width: '))
3  Area = Width * Length
4  print('Area is', Area)
```



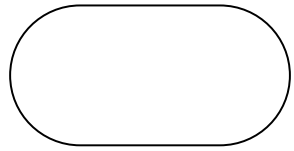
Flowcharts

Flowline



Shows the process's order of operation. A line coming from one symbol and pointing at another.

Terminal



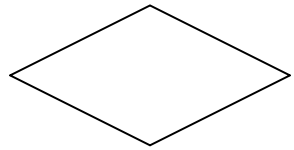
Indicates the beginning and ending of a program or sub-process.

Flowcharts

Process

Represents a set of operations that changes value, form, or location of data.

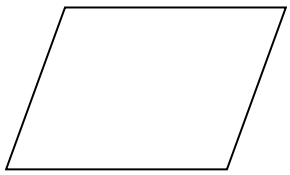
Desision



Shows a conditional operation that determines which one of the two paths the program will take.

Flowcharts

Input/Output

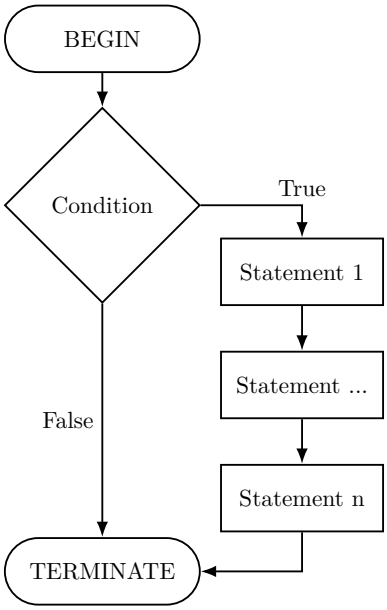


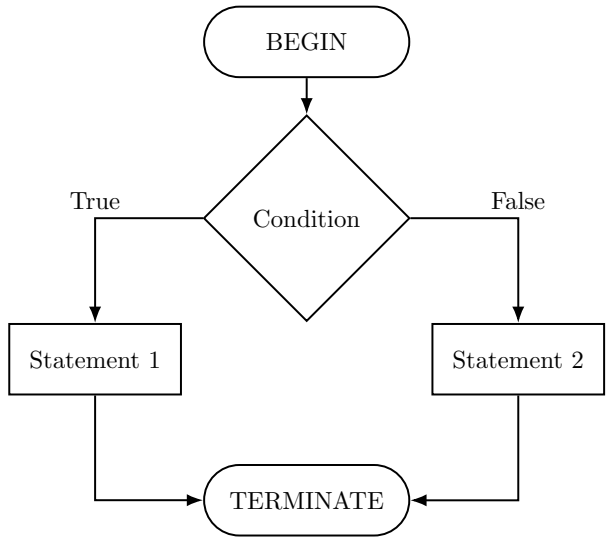
Indicates the process of inputting and outputting data, as in entering data or displaying results.

More ...

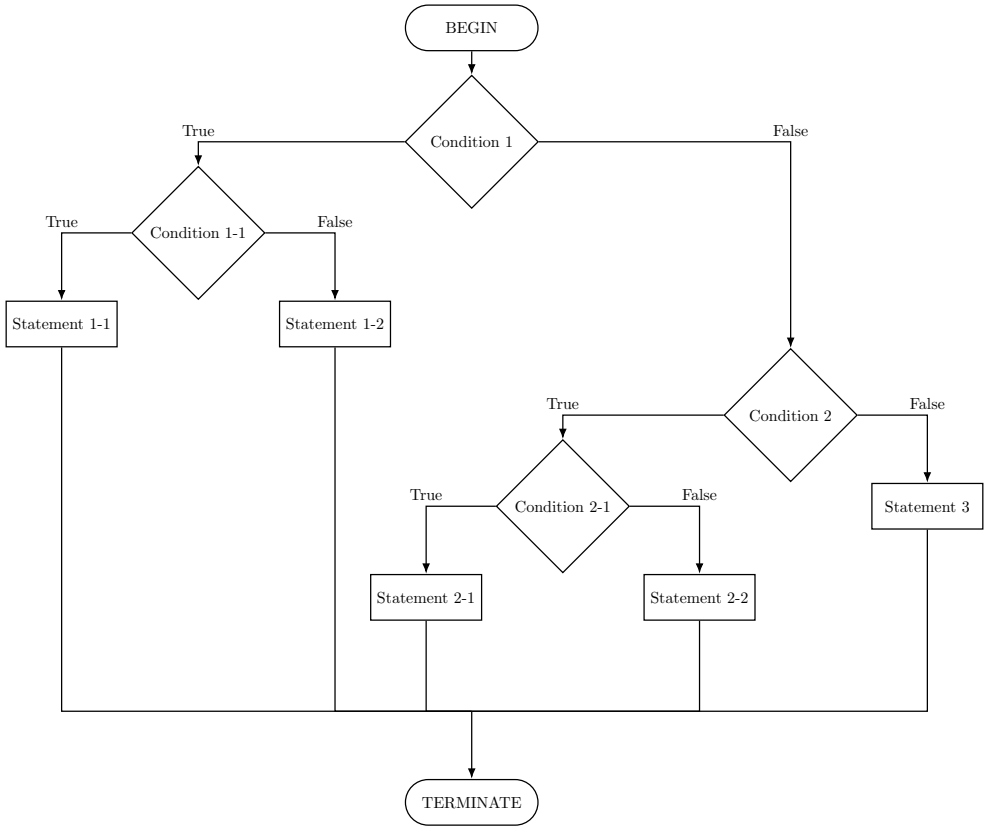
See Chapter 3 of the course materials.

Condition Control Statements





Condition Control Statements

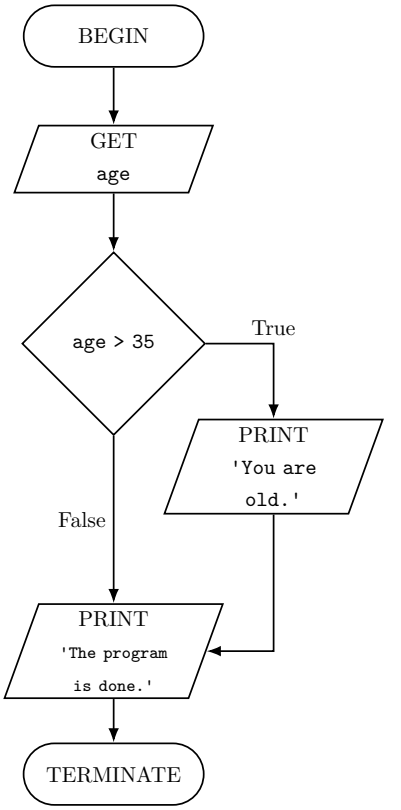


Condition Control Statements

ask_age.py

```
1 age = float(input('How old are you? '))
2 if age > 35:
3     print('You are old.')
4 print('The program is done.')
```

```
>_ python ask_age.py
How old are you? 42
You are old.
The program is done.
```

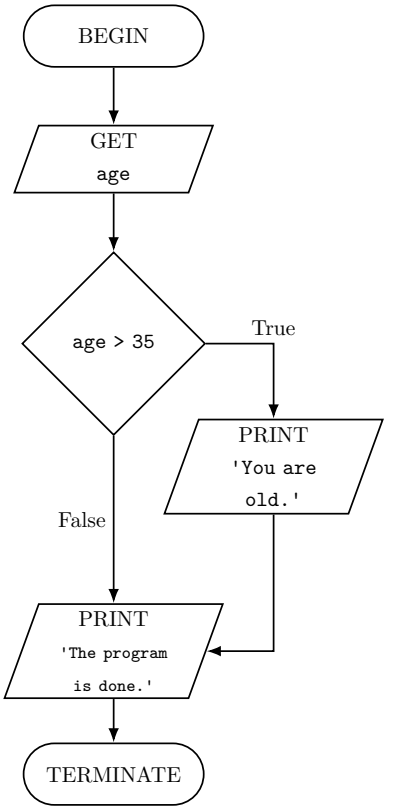


Condition Control Statements

ask_age.py

```
1  age = float(input('How old are you? '))
2  if age > 35:
3      print('You are old.')
4  print('The program is done.')
```

```
>_ python ask_age.py
How old are you? 12
The program is done.
```

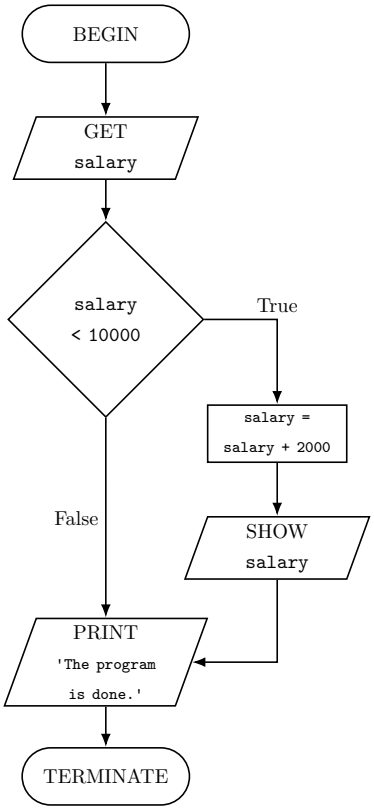


Condition Control Statements

ask_salary.py

```
1 salary = float(input('Enter the salary: '))
2 if salary < 10000:
3     salary = salary + 2000
4     print('Now, your salary is', salary)
5 print('The program is done.')
```

```
>_ python ask_salary.py
Enter the salary: 7500
Now, your salary is 9500
The program is done.
```

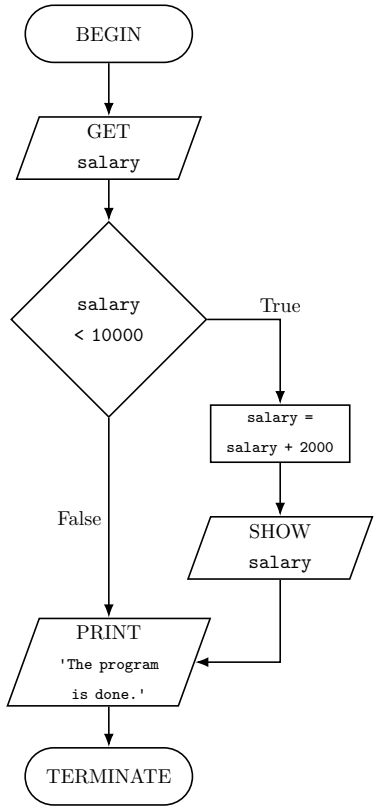


Condition Control Statements

ask_salary.py

```
1 salary = float(input('Enter the salary: '))
2 if salary < 10000:
3     salary = salary + 2000
4     print('Now, your salary is', salary)
5 print('The program is done.')
```

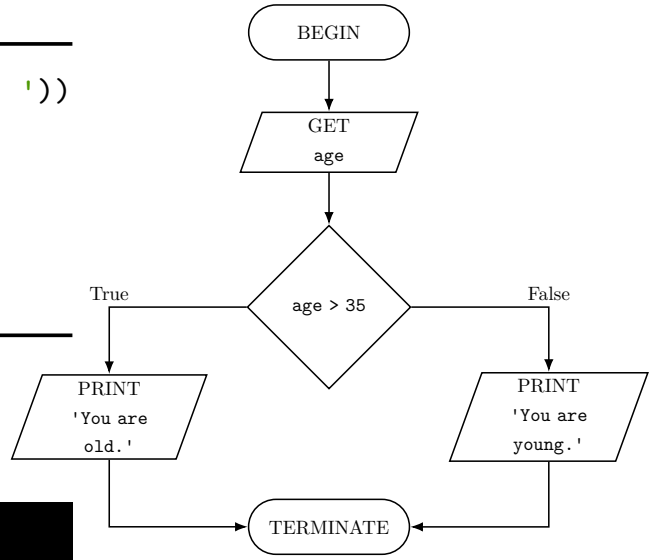
```
>_ python ask_salary.py
Enter the salary: 10000
The program is done.
```



Condition Control Statements

ask_age_v2.py

```
1 age = float(input('How old are you? '))
2 if age > 35:
3     print('You are old.')
4 else:
5     print('You are young.')
```

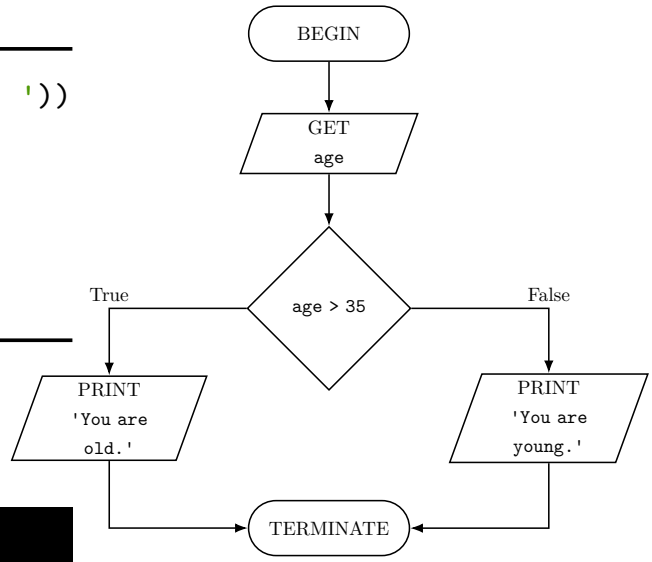



```
>_ python ask_age_v2.py
How old are you? 35
You are young.
```

Condition Control Statements

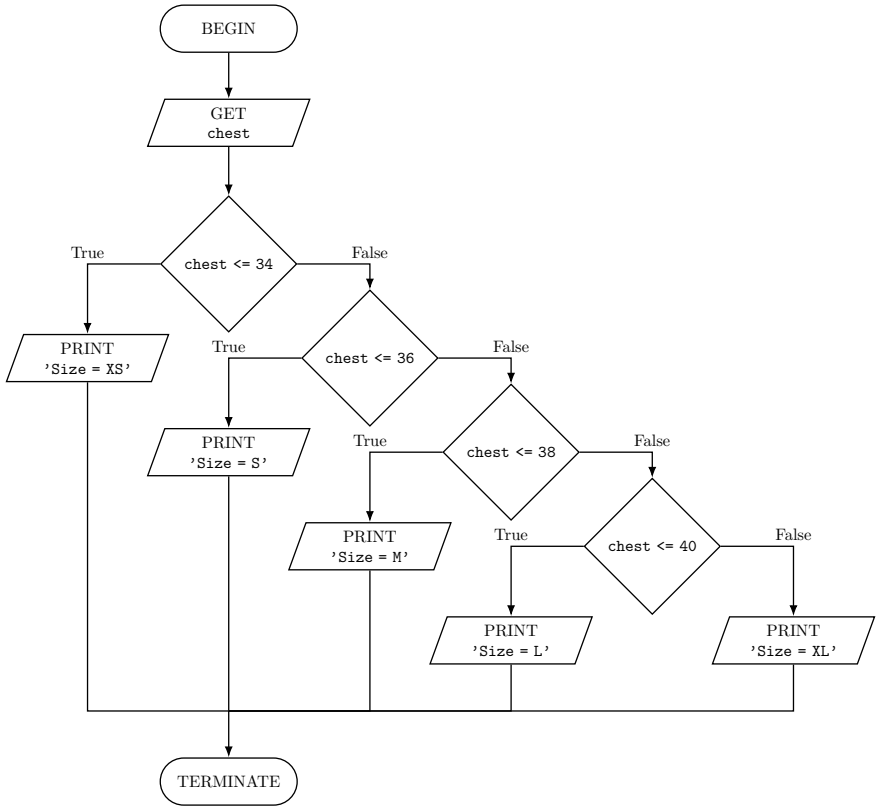
ask_age_v2.py

```
1 age = float(input('How old are you? '))
2 if age > 35:
3     print('You are old.')
4 else:
5     print('You are young.')
```




```
>_ python ask_age_v2.py
How old are you? 53
You are old.
```

Condition Control Statements



Condition Control Statements

ch_size.py

```
1  chest = float(input('Chest length? '))
2  if chest <= 34:
3      print('Size = XS')
4  elif chest <= 36:
5      print('Size = S')
6  elif chest <= 38:
7      print('Size = M')
8  elif chest <= 40:
9      print('Size = L')
10 else:
11     print('Size = XL')
```

```
>_ python ch_size.py
Chest length? 24
Size = XS
```

```
>_ python ch_size.py
Chest length? 37.5
Size = M
```

```
>_ python ch_size.py
Chest length? 46
Size = XL
```

Condition Control Statements

ch_size.py

```
1  chest = float(input('Chest length? '))
2  if chest <= 34:
3      size = 'XS'
4  elif chest <= 36:
5      size = 'S'
6  elif chest <= 38:
7      size = 'M'
8  elif chest <= 40:
9      size = 'L'
10 else:
11     size = 'XL'
12 print('Size =', size)
```

```
>_ python ch_size.py
Chest length? 24
Size = XS
```

```
>_ python ch_size.py
Chest length? 37.5
Size = M
```

```
>_ python ch_size.py
Chest length? 46
Size = XL
```

Exercise

Write a Python program to find the minimum of three values.

min_val.py

```
1  x = float(input('Enter a number: '))
2  y = float(input('Enter a number: '))
3  z = float(input('Enter a number: '))
4  minimum = x
5  if y < minimum:
6      minimum = y
7  if z < minimum:
8      minimum = z
9  print('The minimum is', minimum)
```

```
>_ python min_val.py
Enter a number: 18
Enter a number: 2
Enter a number: 999
The minimum is 2
```

```
>_ python min_val.py
Enter a number: 56.34
Enter a number: 3.14
Enter a number: 0.11
The minimum is 0.11
```

Exercise

Write a Python program to check if the input integer is whether even or odd.

check_int.py

```
1 x = int(input('Enter an integer: '))
2 r = x % 2
3 if r == 0:
4     result = 'even'
5 else:
6     result = 'odd'
7 print(x, 'is', result)
```

```
>_ python check_int.py
Enter an integer: 1444
1444 is even
```

```
>_ python check_int.py
Enter an integer: -361
-361 is odd
```

Exercise

For the first two years, a dog year is equal to 10.5 human years.
After that, each dog year equals 4 human years.
Write a Python program to calculate a dog's age in dog's years.

dog_age.py

```
1 dog_age = float(input('Dog Age: '))
2 if dog_age <= 2:
3     dog_year = 10.5 * dog_age
4 else:
5     dog_year = 10.5*2 + 4*(dog_age-2)
6 print('Dog year is', dog_year)
```

```
>_ python dog_age.py
Dog Age: 15
Dog year is 73
```

Exercise

Write a Python program to translate a score to the corresponding level. Use the following table for the translation.

level.py

```
1  score = float(input('Enter a score: '))
2  if (score < 0) or (score > 100):
3      print('Score is out of range')
4      print('Please try again')
5  else:
6      if score < 60:
7          level = 'NOOB'
8      elif score < 80:
9          level = 'SUPER'
10     else:
11         level = 'LEGEND'
12     print('Your level is', level)
```

Score	Level
[0, 60)	NOOB
[60, 80)	SUPER
[80, 100]	LEGEND

```
>_ python level.py
Enter a score: 99
Your level is LEGEND
```

```
>_ python level.py
Enter a score: 999
Score is out of range
Please try again
```