Chương 6. Đầu vào người dùng và vòng lặp

NGUYỄN HOÀNG ANH

CNTT1. PTIT

Nội dung

Cách hàm Input() hoạt động

Giới thiệu vòng lặp while

Sử dụng vòng lặp while với danh sách và từ điển

6.1. Cách hàm Input() hoạt động

Hàm input() tạm dừng chương trình của chúng ta và đợi người dùng nhập một số tiếp theo. Khi Python nhận được đầu vào của người dùng, nó sẽ gán đầu vào đó cho biến để giúp thuận tiện khi làm việc.

```
message = input("Tell me something, and I will repeat it back to you: ")
print(message)
```

Hàm input() nhận một đối số: dấu nhắc hoặc hướng dẫn, mà chúng ta muốn hiển thị cho người dùng để họ biết phải làm gì. Trong ví dụ này, khi Python chạy dòng đầu tiên, người dùng sẽ thấy lời nhắc “Tell me something, and I will repeat it back to you: ".

Chương trình đợi trong khi người dùng nhập phản hồi của họ và tiếp tục sau khi người dùng nhấn enTer. Câu trả lời là được gán cho biến message, sau đó hàm print (message) hiển thị tại đầu vào cho người dùng:

```
Tell me something, and I will repeat it back to you: Hello everyone!
Hello everyone!
```

Viết lời nhắc rõ ràng

Mỗi khi sử dụng hàm input(), ta nên bao gồm một lời nhắc rõ ràng, dễ theo dõi cho người dùng biết chính xác loại thông tin chương trình đang tìm kiếm. Bất kỳ câu lệnh nào cho người dùng biết những gì cần nhập. Ví dụ:

```
name = input("Please enter your name: ")
print(f"\nHello, {name}!")
```

```
Please enter your name: Eric
Hello, Eric!
```

viết lời nhắc dài hơn một dòng

Viết lời nhắc rõ ràng

Viết lời nhắc dài hơn một dòng: Gán lời nhắc của mình cho một biến và chuyển biến đó vào hàm input()

```
prompt = "If you tell us who you are, we can personalize the messages you see."
prompt += "\nWhat is your first name? "
name = input(prompt)
print(f"\nHello, {name}!")
```

```
If you tell us who you are, we can personalize the messages
you see.
What is your first name? Eric

Hello, Eric!
```

Sử dụng int() để nhận đầu vào số nguyên

Khi ta sử dụng hàm input(), Python sẽ thông dịch mọi thứ mà người dùng nhập dưới dạng một chuỗi. Cùng xem xét phiên thông dịch viên sau đây, phiên này yêu cầu tuổi của người dùng:

```
>>> age = input("How old are you? ")
How old are you? 21
>>> age
'21'
```

Nếu chỉ yêu cầu là in đầu vào thì cách này hoạt động tốt. Nhưng nếu ta cố gắng sử dụng đầu vào là một số, ta sẽ gặp lỗi:

```
>>> age = input("How old are you? ")
How old are you? 21
①   >>> age >= 18
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
②       TypeError: unorderable types: str() >= int()
```

Sử dụng int() để nhận đầu vào số nguyên

Chúng ta có thể giải quyết vấn đề này bằng cách sử dụng hàm int(), cho biết Python để coi đầu vào là một giá trị số. Hàm int() chuyển đổi biểu diễn chuỗi của một số thành biểu diễn số

```
>>> age = input("How old are you? ")
How old are you? 21
①   >>> age = int(age)
>>> age >= 18
True
```

Sử dụng int() để nhận đầu vào số nguyên

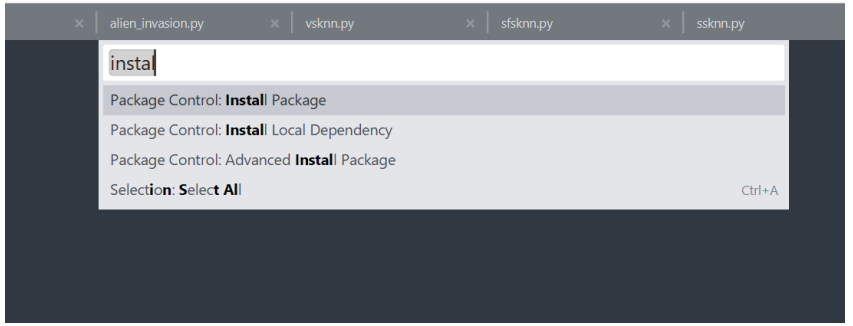
Sử dụng hàm int() trong một chương trình thực tế

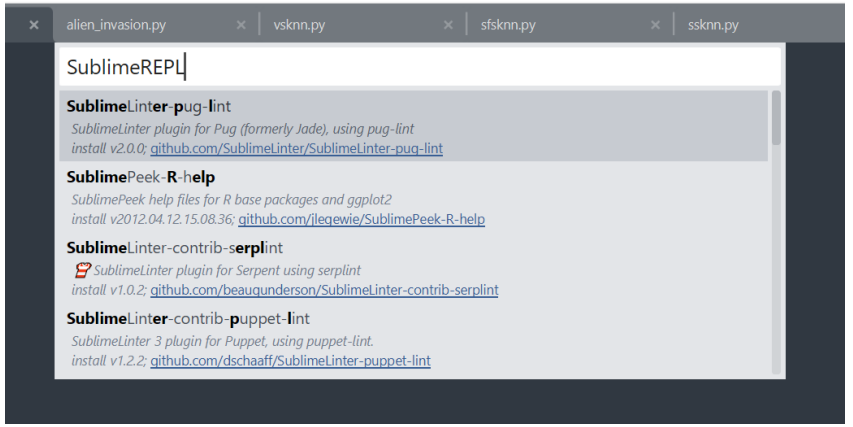
```
height = input("How tall are you, in inches? ")

height = int(height)

if height >= 48:

    print("\nYou're tall enough to ride!")

else:

    print("\nYou'll be able to ride when you're a little older.")
```

```
How tall are you, in inches? 71

You're tall enough to ride!
```

Khắc phục lỗi không nhận input() với Sublime Text

1. open Sublime Text >> CTRL + P
2. CTRL + P will open the Package control
3. Click on Package Control: Install package
4. Wait for a sec to pop up a search bar.
5. Type SublimeREPL and Click it. It'll get installed in a few secs.

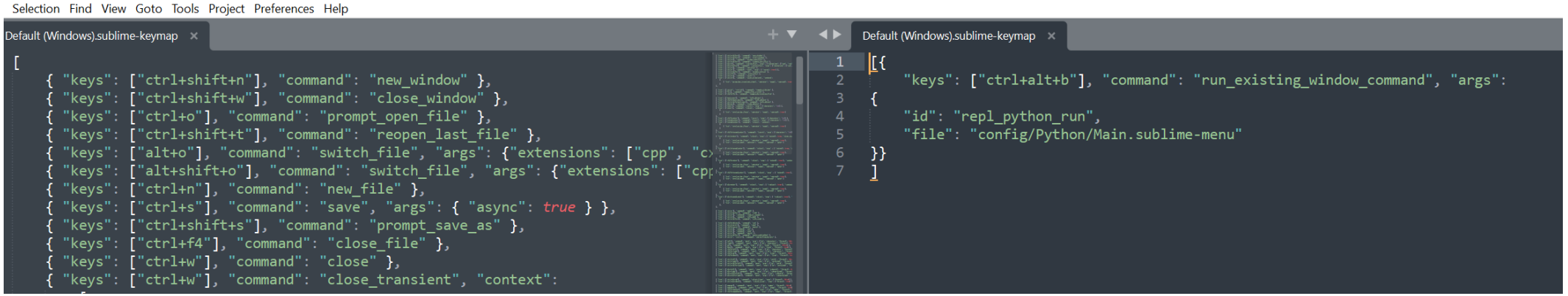




Then follow the following steps to run your program;
6.Tools >> SublimeREPL >> Python >> Python run Current File

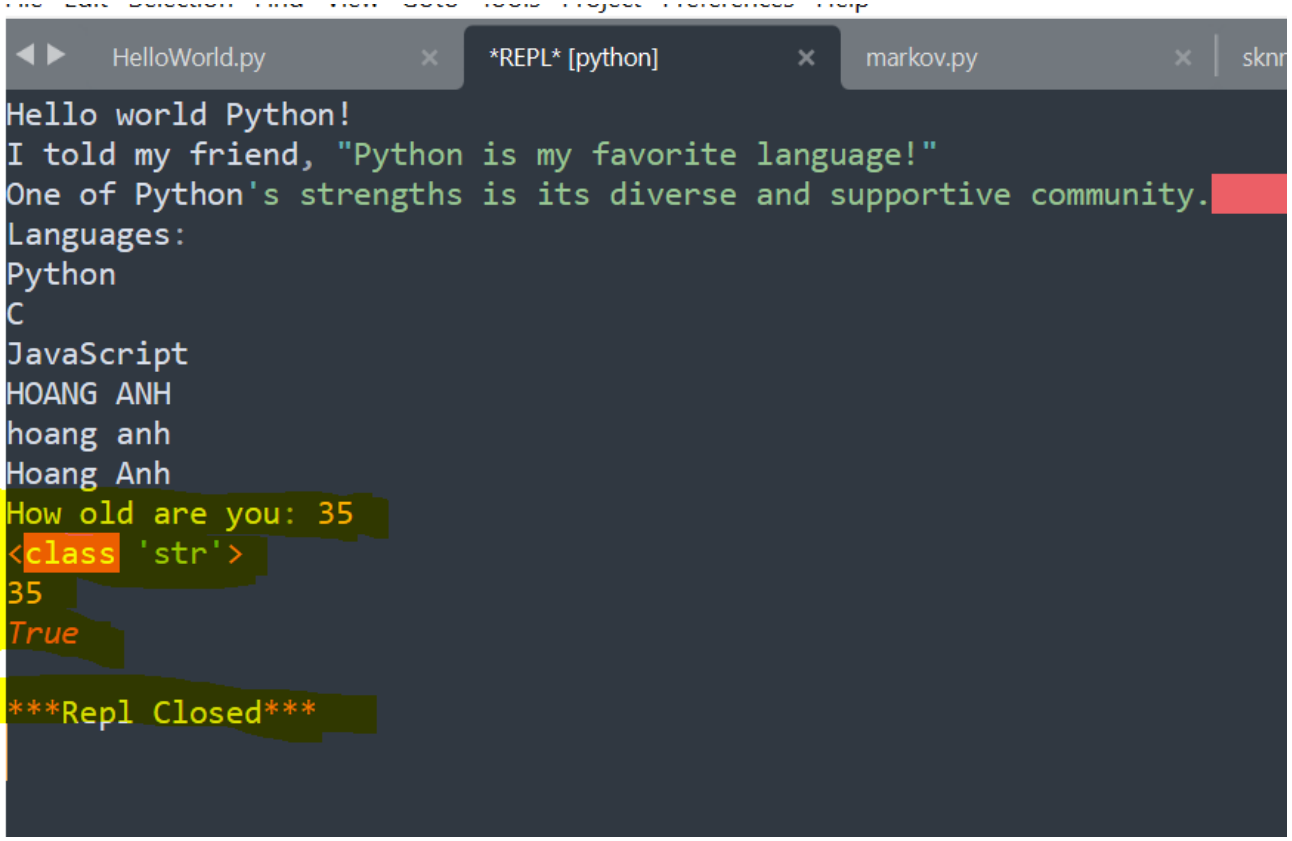
shortKey: ctrl+alt+b

Go to Preferences->Keys Bindings




```
[{
      "keys": ["ctrl+alt+b"], "command": "run_existing_window_command", "args":
{
   "id": "repl_python_run",
   "file": "config/Python/Main.sublime-menu"
}}
]
```

Kết quả:


HelloWorld.py
REPL [python]
markov.py
Hello world Python!
I told my friend, "Python is my favorite language!"
One of Python's strengths is its diverse and supportive community.
Languages:
Python
C
JavaScript
HOANG ANH
hoang anh
Hoang Anh
How old are you: 35
<class 'str'>
35
True
Repl Closed

Toán tử modulo

Một công cụ hữu ích để làm việc với thông tin số là toán tử modulo(%), chia một số cho một số khác và trả về phần còn lại.

Toán tử mô-đun không cho biết một số gấp bao nhiêu lần số khác; nó chỉ cho ta biết phần còn lại là bao nhiêu

Khi một số chia hết cho một số khác thì số dư là 0, vì vậy toán tử modulo luôn trả về 0. Ta có thể sử dụng dữ kiện này để xác định nếu một số chẵn hoặc lẻ

```
>>> 4 % 3
1
>>> 5 % 3
2
>>> 6 % 3
0
>>> 7 % 3
1
```

```
number = input("Enter a number, and I'll tell you if
it's even or odd: ")
number = int(number)
if number % 2 == 0:
    print(f"\nThe number {number} is even.")
else:
    print(f"\nThe number {number} is odd.")
```

```
Enter a number, and I'll tell you if it's even or odd: 42
The number 42 is even.
```

6.2. Vòng lặp while

Vòng lặp for lấy một tập hợp các mục và thực thi một khối mã một lần cho mỗi mục trong tập hợp. Ngược lại, vòng lặp while chạy *miễn là*, hoặc *trong khi*, một điều kiện nhất định là đúng.

Vòng lặp với hành động

Có thể sử dụng vòng lặp while để đếm dần một chuỗi số. Ví dụ, vòng lặp while sau đếm từ 1 đến 5

```
current_number = 1
while current_number <= 5:
    print(current_number)
    current_number += 1
```

```
1
2
3
4
5
```

Python lặp lại vòng lặp miễn là điều kiện current_number <= 5 là đúng

Các chương trình ta sử dụng hàng ngày rất có thể chứa các vòng lặp while. Ví dụ, một trò chơi cần một vòng lặp while khi tiếp tục chạy khi ta muốn và nó có thể ngừng chạy ngay sau khi ta yêu cầu nó thoát.

Để người dùng lựa chọn khi nào thoát

Xây dựng chương trình parrot.py để người dùng muốn luôn chạy bằng cách đặt hầu hết chương trình bên trong vòng lặp while

Xác định một giá trị thoát và sau đó giữ chương trình chạy miễn là người dùng chưa nhập giá trị thoát

```
①   prompt = "\nTell me something, and I will repeat it back to you:"
    prompt += "\nEnter 'quit' to end the program. ”
②   message = ""
③   while message != 'quit':
        message = input(prompt)
        print(message)
```

Để người dùng lựa chọn khi nào thoát

Tại message = input(prompt), Python hiển thị lời nhắc và đợi người dùng nhập thông tin đầu vào của họ. Bất cứ thứ gì người dùng nhập đều được gán vào biến message và in ra; sau đó, Python đánh giá lại điều kiện của vòng lặp while.

Miễn là người dùng chưa nhập từ ‘quit’, lời nhắc được hiển thị lại và Python chờ thêm đầu vào. Khi người dùng cuối cùng nhập ‘quit’, Python dừng thực hiện vòng lặp while và chương trình kết thúc:

```
Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. Hello everyone!
Hello everyone!

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. Hello again.
Hello again.

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. quit
quit
```

Để người dùng lựa chọn khi nào thoát

Chương trình này hoạt động tốt, ngoại trừ việc nó in từ ‘quit’ như thể nó là một thông điệp thực tế. Một kiểm tra có điều kiện đơn giản nếu giải quyết được điều này

```
prompt = "\nTell me something, and I will repeat it back to you:"
prompt += "\nEnter 'quit' to end the program. "
message = ""
while message != 'quit':
    message = input(prompt)
    if message != 'quit':
        print(message)
```

```
Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. Hello everyone!
Hello everyone!

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. Hello again.
Hello again.

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. Quit
```

Sử dụng Flag (cờ)

Đối với một chương trình sẽ chạy miễn là có nhiều điều kiện đúng, ta có thể xác định một biến xác định xem toàn bộ chương trình có đang hoạt động hay không. Biến này, được gọi là flag, hoạt động như một tín hiệu cho chương trình

```
①   active = True
②   while active:
        message = input(prompt)
③       if message == 'quit':
            active = False
④       else:
            print(message)
```

Chương trình này có đầu ra giống như ví dụ trước, nơi chúng ta đã đặt kiểm tra điều kiện trực tiếp trong câu lệnh while Nhưng bây giờ chúng ta có một flag để cho biết liệu chương trình tổng thể có ở trạng thái hoạt động hay không, nó sẽ dễ dàng thêm nhiều kiểm tra hơn cho các sự nên làm cho *active* trở thành False

Sử dụng **break** để thoát khỏi vòng lặp

Để thoát khỏi vòng lặp while ngay lập tức mà không cần chạy bất kỳ mã nào còn lại trong vòng lặp, bất kể kết quả của bất kỳ kiểm tra điều kiện nào, hãy sử dụng câu lệnh break

```
prompt = "\nPlease enter the name of a city you have visited:"
prompt += "\n(Enter 'quit' when you are finished.) "
while True:
    city = input(prompt)
    if city == 'quit':
        break
    else:
        print(f"I'd love to go to {city.title()}!")
```

```
Please enter the name of a city you have visited:
(Enter 'quit' when you are finished.) Hanoi
I'd love to go to Hanoi !

Please enter the name of a city you have visited:
(Enter 'quit' when you are finished.) Saigon
I'd love to go to Saigon!

Please enter the name of a city you have visited:
(Enter 'quit' when you are finished.) quit
```

Sử dụng **continue** trong vòng lặp

Thay vì thoát ra khỏi vòng lặp hoàn toàn mà không thực hiện phần còn lại của nó , ta có thể sử dụng câu lệnh continue để quay lại phần đầu của vòng lặp dựa trên kết quả của một bài kiểm tra có điều kiện

```
current_number = 0                          1
while current_number < 10:                  3
    current_number += 1                     5
    if current_number % 2 == 0:             7
        continue                            9
    print(current_number)
```

Tránh lặp vô hạn

Mỗi vòng lặp while cần một cách để dừng chạy để nó sẽ không tiếp tục chạy mãi mãi.

```
#normal
x = 1
while x <= 5:
    print(x)
    x += 1
```

```
# This loop runs forever!
x = 1
while x <= 5:
    print(x)
```

```
1
1
1
1
--snip--
```

Nếu chương trình bị mắc kẹt trong một vòng lặp vô hạn, hãy nhấn cTrL-C hoặc chỉ cần đóng cửa sổ đầu cuối hiển thị đầu ra chương trình

Đảm bảo ít nhất một phần của chương trình có thể làm cho điều kiện của vòng lặp True hoặc chương trình tới được một câu lệnh break.

6.3. Sử dụng vòng lặp while với danh sách và từ điển

Để theo dõi nhiều người dùng và các thông tin, chúng ta sẽ cần sử dụng danh sách và từ điển với vòng lặp while.

Vòng lặp for có hiệu quả để lặp qua một danh sách, nhưng chúng ta không nên sửa đổi danh sách bên trong vòng lặp for vì Python sẽ gặp khó khăn trong việc theo dõi các phần tử trong danh sách. Để sửa đổi danh sách khi ta làm việc với nó, hãy sử dụng vòng lặp while. Sử dụng vòng lặp while với danh sách và từ điển cho phép ta thu thập, lưu trữ và tổ chức nhiều đầu vào để kiểm tra và báo cáo về sau.

Chuyển phần tử từ danh sách này sang danh sách khác

Sử dụng vòng lặp while để kéo người dùng khỏi danh sách người dùng chưa được xác minh và sau đó thêm họ vào một danh sách riêng người dùng đã xác minh

```
unconfirmed_users = ['alice', 'brian', 'candace']
confirmed_users = []
while unconfirmed_users:
    current_user = unconfirmed_users.pop()
    print(f"Verifying user: {current_user.title()}")
    confirmed_users.append(current_user)
# Display all confirmed users.
print("\nThe following users have been confirmed:")
for confirmed_user in confirmed_users:
    print(confirmed_user.title())
```

```
Verifying user: Candace
Verifying user: Brian
Verifying user: Alice

The following users have been confirmed:
Candace
Brian
Alice
```

Xóa tất cả các thể hiện của một giá trị trong một danh sách

Giả sử ta có một danh sách các vật nuôi có giá trị 'cat' được lặp lại nhiều lần. Để loại bỏ tất cả các trường hợp của giá trị đó, ta có thể chạy một vòng lặp while cho đến khi 'cat' không còn còn trong danh sách

```
pets = ['dog', 'cat', 'dog', 'goldfish', 'cat', 'rabbit', 'cat']
print(pets)
while 'cat' in pets:
    pets.remove('cat')
print(pets)
```

```
['dog', 'cat', 'dog', 'goldfish', 'cat', 'rabbit', 'cat']
['dog', 'dog', 'goldfish', 'rabbit']
```

Điền từ điển với đầu vào người dùng

```
responses = {}
# Set a flag to indicate that polling is active.
polling_active = True
while polling_active:
# Prompt for the person's name and response.
    name = input("\nWhat is your name? ")
    response = input("Which mountain would you like to
climb someday? ")
# Store the response in the dictionary.
    responses[name] = response
# Find out if anyone else is going to take the poll.
    repeat = input("Would you like to let another person
respond? (yes/ no) ")
    if repeat == 'no':
        polling_active = False
# Polling is complete. Show the results.
print("\n--- Poll Results ---")
for name, response in responses.items():
    print(f"{name} would like to climb {response}.")
```

```
What is your name? HoangAnh
Which mountain would you like to climb someday? Ba Vi
Would you like to let another person respond? (yes/ no) yes

What is your name? Quang Minh
Which mountain would you like to climb someday? Tam Dao
Would you like to let another person respond? (yes/ no) yes

What is your name? Quynh Anh
Which mountain would you like to climb someday? Bana
Would you like to let another person respond? (yes/ no) no

--- Poll Results ---
HoangAnh would like to climb Ba Vi.
Quang Minh  would like to climb Tam Dao .
Quynh Anh would like to climb Bana.

***Repl Closed***
```

Kết chương

Trong chương này, chúng ta đã học cách sử dụng input() để cho phép người dùng cung cấp thông tin riêng của họ trong các chương trình của mình. Ta đã học cách làm việc với cả hai đầu vào văn bản và số và cách sử dụng vòng lặp while để khiến chương trình chạy cho tới khi người dùng muốn thoát. Chúng đã thấy một số cách để kiểm soát luồng thực hiện một vòng lặp while bằng cách đặt một giá trị active flag, sử dụng câu lệnh break, và bằng cách sử dụng câu lệnh continue.

Chúng ta đã học cách sử dụng vòng lặp while để di chuyển các phần tử từ danh sách này sang danh sách khác và cách xóa tất cả các thể hiện của một giá trị từ một danh sách. Ta cũng đã biết cách sử dụng vòng lặp while với từ điển.

Trong Chương 7, chúng ta sẽ tìm hiểu về các hàm. Các hàm cho phép ta chia các chương trình thành các phần nhỏ, mỗi phần thực hiện một công việc cụ thể. Ta có thể gọi một hàm bao nhiêu lần tùy thích và ta có thể lưu trữ các hàm trong các tệp riêng biệt. Bằng cách sử dụng các hàm, ta sẽ có thể viết nhiều hơn mã hiệu quả giúp khắc phục sự cố và duy trì dễ dàng hơn và điều đó có thể được sử dụng lại trong nhiều chương trình khác nhau

Bài tập

6-1. Rental Car: Viết một chương trình hỏi người dùng loại xe nào họ muốn thuê, sau đó viết ra một thông điệp về loại xe đó. Ví dụ: “Let me see if I can fnd you a Subaru.”

6-2. Restaurant Seating: viết một chương trình hỏi người dùng muốn đặt bàn có bao nhiêu chỗ. Nếu câu trả lời lớn hơn 8, in ra thông điệp là họ cần chờ để sắp xếp bàn. Nếu không, in ra thông điệp là bàn họ đặt đã sẵn sàng.

6-3. Multiples of Ten: viết chương trình yêu cầu nhập vào một số, sau đó in ra câu trả lời là số đó có phải là bội của 10 hay không.

6-4. Pizza Toppings: viết một chương trình có vòng lặp yêu cầu người dùng nhập vào tên các lớp phủ bánh pizza (toppings) cho tới khi người dùng gõ vào giá trị ‘quit’. Với mỗi giá trị lớp phủ bánh người dùng đã thêm, in ra một thông điệp rằng ta đã thêm lớp phủ đó vào bánh cho họ.

6-5. Movie Tickets: một rạp chiếu phim bán vé có giá thay đổi dựa trên tuổi của khách hàng. Nếu khách hàng dưới 3 tuổi – miễn phí vé; Nếu khách hàng từ 3 tới 12 tuổi – giá vé là 10$; Nếu khách hàng lớn hơn 12 tuổi – giá vé là 15$. Viết chương trình yêu cầu khách hàng nhập vào tuổi, sau đó in ra giá vé cho người dùng.

Bài tập

6-6. Three Exits: làm các ví dụ 6-4, 6-5 với các cách sau, mỗi cái ít nhất 1 lần:

- sử dụng kiểm tra có điều kiện trong câu lệnh while để dừng vòng lặp

- sử dụng một biến active để điều khiển số lượt chạy của vòng lặp

- sử dụng break để thoát vòng lặp khi người dùng gõ giá trị ‘quit’

6-7. Infinity: viết một vòng lặp không kết thúc và chạy nó. (Để kết thúc vòng lặp nhấn tổ hợp ctrl-C hoặc đóng cửa sổ đang hiển thị đầu ra).

6-8. Deli: Tạo một danh sách có tên là sandwich_orders và điền vào đó tên của các loại bánh sandwich khác nhau. Sau đó, tạo một danh sách trống được gọi là finish_sandwiches. Lặp lại danh sách các đơn đặt hàng bánh sandwich và in thông báo cho mỗi đơn đặt hàng, chẳng hạn như tôi đã làm bánh mì cá ngừ cho bạn. Khi mỗi chiếc bánh sandwich được làm xong, hãy chuyển nó vào danh sách finish_sandwiches. Sau khi tất cả các loại bánh mì đã được làm xong, hãy in một thông báo liệt kê từng loại bánh sandwich đã được làm.

6-9. No Pastrami: Giả sử trong danh sách bài 6-8 có 3 cái thuộc loại pastrami. Tại phần đầu của mã, in ra là cửa hàng hết pastrami và sử dụng vòng lặp while để loại bỏ tất cả các bánh sandwich loại pastrami trong sandwich_orders. In ra danh sách sandwich_orders để đảm bảo đã loại bỏ hết các sandwich loại pastrami.

6-10 Dream Vacation: Viết một chương trình thăm dò ý kiến người dùng về kỳ nghỉ mơ ước của họ. Viết lời nhắc tương tự như Nếu bạn có thể đến thăm một nơi trên thế giới, bạn sẽ đi đâu? Bao gồm một khối mã in kết quả của cuộc thăm dò ý kiến.