

NHẬP MÔN LẬP TRÌNH (23CLC05)

Buổi 4 – 23/10/23

Đâu là sức mạnh thực sự của máy tính?

Ngôn ngữ tự nhiên	C/C++
Nói 100 lần câu sau đây: "I love you!".	<pre>#include <iostream> using namespace std; int main() { for(int i = 1; i <= 100; i++) cout << "I love you!" << endl; }</pre>
Python	
<pre>for i in range(100): print("I \u2764 you!")</pre>	

Cấu trúc lặp là gì?

- Khả năng làm “lặp lại nhiều lần một việc đơn giản” là sở trường và sức mạnh của máy tính
- **Cấu trúc lặp** (iteration structure, looping structure) mô tả việc lặp lại một công việc theo cách nào đó
- Các cấu trúc lặp hay gặp là
 - “lặp một số lần xác định”
 - “lặp theo dãy các giá trị”
 - “lặp trong khi”
 - “lặp cho đến khi”

Lệnh lặp là gì?

- Các ngôn ngữ lập trình cung cấp các **lệnh lặp** (iteration statement, looping statement) khác nhau để hiện thực các cấu trúc lặp
- Các lệnh lặp trong Python là for, while
- Các lệnh lặp trong C/C++ là for, while, do-while
- **Thân lặp** (looping body), là công việc được lặp lại, là một khối lệnh
 - (nhớ lại, trong C/C++, khối lệnh gồm chỉ 1 lệnh có thể không cần cặp ngoặc nhọn { })
- **Điều kiện lặp** (looping condition) trong một số cấu trúc lặp là biểu thức luận lý

Cộng điểm

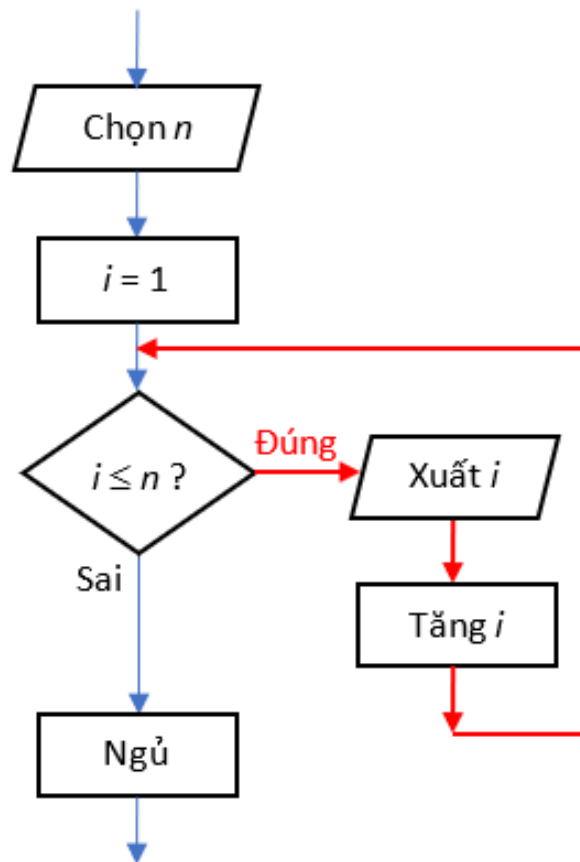
- Viết chương trình “tự khai căn” (Bài tập 1.4.3 - Tài liệu C)

Làm sao “tham số hóa” công việc cần lặp lại?

Lặp lại việc sau đây với i lần lượt nhận các giá trị 1, 2, 3, ..., n :

- Đếm con cừu i

Lưu đồ



`import time`

`n = int(input("Đếm tới mấy? "))`

`for i in range(1, n + 1):`

`print(i, "con cừu"); time.sleep(1.0)`

`print(i, "con cừu"); time.sleep(1.0)`

`print("Ngủ đi!")`

C/C++

`#include <iostream>`

`using namespace std;`

`int main() {`

`int n;`

`printf("Dem toi may? ");`

`cin >> n;`

`for(int i = 1; i <= n; i++)`

`{`

`cout << i << " con cuu" << endl;`

`cout << i << " con cuu" << endl;`

`}`

`cout << "Ngu di!" << endl;`

`}`

Tham số hóa công việc là gì?

- **Tham số hóa** (parameterization) công việc là việc tách bạch phần cố định S và phần thay đổi i của công việc
 - Khi cho **tham số** (parameter) i nhận giá trị cụ thể, ta có công việc cụ thể $S(i)$
- Trong ví dụ trên, ta tham số hóa công việc “xuất số i ” với phần cố định S là “xuất số” và tham số i là “số cần xuất”
- Để lặp lại một công việc được tham số hóa, ta dùng **biến lặp** (loop variable) cho tham số và thân lặp thực hiện phần cố định theo biến lặp
- Lệnh for điều khiển việc cho biến lặp lần lượt nhận các giá trị trong dãy giá trị nào đó và thực hiện thân lặp theo mỗi giá trị này

Cộng điểm

- Làm bài tập 1.5.5b (Tài liệu C)

Giải lao!!!

- Giải lao đến 9h50

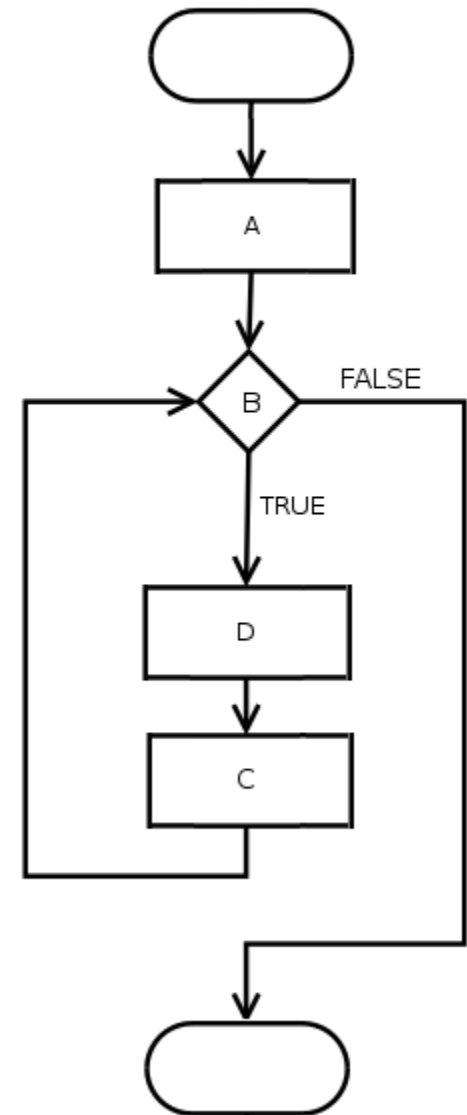
Có thể điều khiển biến lặp không?

Python (cách 1)	Python (cách 2)
<pre>n = int(input("n = ? ")) for i in range(1, n + 1, 2): print(i)</pre>	<pre>n = int(input("n = ? ")) for i in range(1, n + 1): if i % 2 == 1: # i lẻ print(i)</pre>
C/C++ (cách 1)	C/C++ (cách 2)
<pre>#include <iostream> using namespace std; int main() { int n; printf("n = ? "); cin >> n; for(int i = 1; i <= n; i += 2) cout << i << endl; }</pre>	<pre>#include <iostream> using namespace std; int main() { int n; printf("n = ? "); cin >> n; for(int i = 1; i <= n; i++) if(i % 2 == 1) // i lẻ cout << i << endl; }</pre>

Chi tiết hơn về lệnh for!

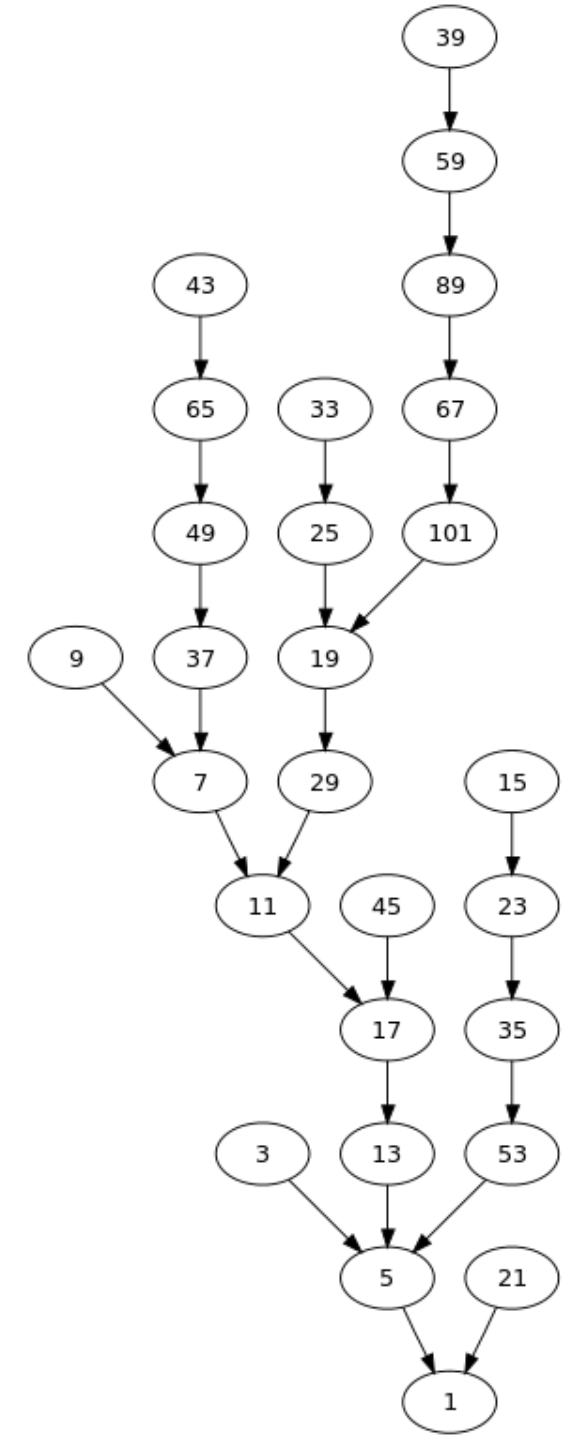
- Python: tra cứu hàm range (https://www.w3schools.com/python/ref_func_range.asp)
- C/C++: nghiên cứu lưu đồ của lệnh for (https://en.wikipedia.org/wiki/For_loop)
- Một cách điều khiển chi tiết lệnh lặp là “lặp đơn giản với lệnh chọn bên trong”

```
for(A;B;C)  
D;
```



Nghi vấn $3n + 1$ là gì?

- Xuất phát từ số nguyên dương n , tính dãy số theo qui tắc
 - Nếu số trước là chẵn thì số sau bằng nửa số trước
 - Nếu số trước là lẻ thì số sau bằng 3 lần số trước cộng thêm 1
- Ví dụ: 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, ...
- **Nghi vấn Collatz:** với mọi giá trị xuất phát n , dãy số tính như trên đều dẫn đến số 1?
- https://en.wikipedia.org/wiki/Collatz_conjecture



Lặp không biết trước số lần lặp! (1)

Lưu đồ	Mã giả
<pre>graph TD; Start([Nhập n]) --> Cond1{n = 1?}; Cond1 -- Đúng --> Exit1[]; Cond1 -- Sai --> Cond2{n chẵn?}; Cond2 -- Đúng --> Proc1[n = n / 2]; Cond2 -- Sai --> Proc2[n = 3n + 1]; Proc1 --> Cond1; Proc2 --> Cond1;</pre>	Cho n là một số nguyên dương, lặp lại quá trình sau đây cho đến khi n là 1 thì dừng: <ul style="list-style-type: none">- Nếu n chẵn: $n = n / 2$,- Nếu n lẻ: $n = 3n + 1$.
	Python (while)
	<pre>n = int(input("Nhập n > 1: ")) while n != 1: print(n, end=" ") if n % 2 == 0: # n chẵn n = n // 2 else: # n lẻ n = 3*n + 1 print(n, end=" ") # n là 1</pre>

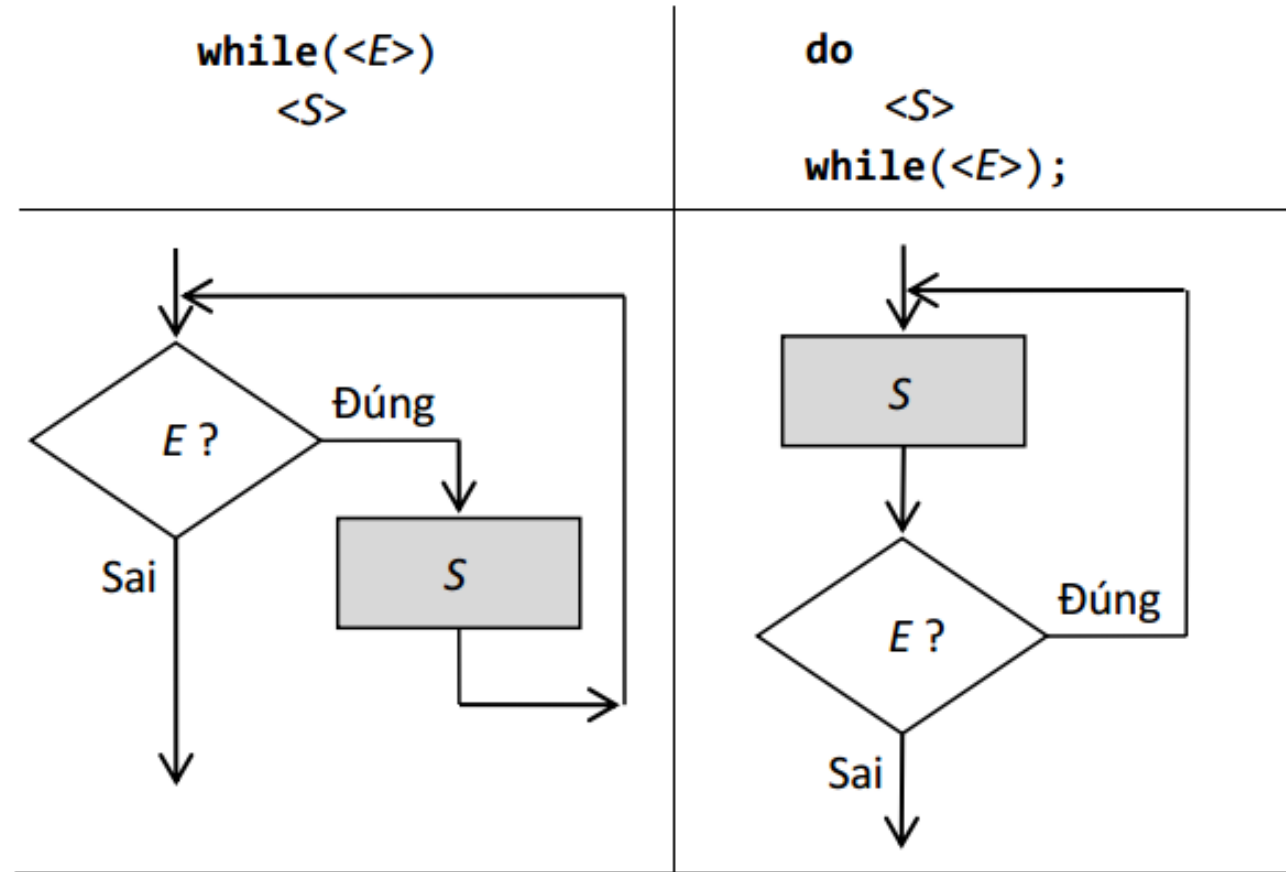
Lặp không biết trước số lần lặp! (2)

C/C++ (do-while)	C/C++ (while)
<pre>#include <iostream> using namespace std; int main() { int n; printf("Nhap n > 1: "); cin >> n; do { cout << n << " "; if(n % 2 == 0) // n chan n = n / 2; else // n le n = 3 * n + 1; } while(n != 1); cout << n << " "; // n la 1 }</pre>	<pre>#include <iostream> using namespace std; int main() { int n; printf("Nhap n > 1: "); cin >> n; while(n != 1) { cout << n << " "; if(n % 2 == 0) // n chan n = n / 2; else // n le n = 3 * n + 1; } cout << n << " "; // n la 1 }</pre>

Khác biệt giữa lệnh lặp while, do-while và for là gì?

- Lệnh while thường được dùng khi rất khó điều khiển biến lặp (hoặc “không có biến lặp” rõ ràng, hoặc “nhiều biến lặp”, ...)
- Trường hợp lặp số lần xác định hoặc biến lặp đơn giản thì nên dùng lệnh for
- C/C++ phân biệt 2 cấu trúc lặp
 - WHILE-DO, kiểm tra điều kiện lặp trước khi thực thi thân lặp, với lệnh while
 - DO-WHILE, thực thi thân lặp trước kiểm tra điều kiện lặp, với lệnh do-while
- Nên dùng lệnh do-while khi thân lệnh được lặp lại ít nhất 1 lần

Khác biệt giữa lệnh lặp while, do-while và for là gì?



Khác biệt giữa lệnh lặp while, do-while và for là gì?

Mã C	Lưu đồ
<pre>for(<Init>; <E>; <Post>) <S> <Init>; while(<E>) { <S> <Post>; }</pre>	<pre>graph TD Entry(()) --> Init[Init] Init --> E{E?} E -- Sai --> Exit(()) E -- Đúng --> S[S] S --> Post[Post] Post --> E</pre>

Còn lệnh lặp REPEAT-UNTIL của Pascal thì sao?

Mã Pascal	Lưu đồ
<pre>REPEAT <S> UNTIL <E>;</pre>	<pre>graph TD; Entry(()) --> S[S]; S --> E{E?}; E -- Sai --> Entry; E -- Đúng --> Exit(());</pre>
<p>Mã C</p> <pre>do <S> while(!<E>;</pre>	

Cộng điểm

- Làm bài tập 1.5.7 (Tài liệu C)

Số nguyên tố là gì?

Tiếng Việt	Python
Cho số nguyên $n > 1$, n được gọi là số nguyên tố (prime) nếu n có đúng 2 ước số không âm là 1 và n , ngược lại n được gọi là hợp số (composite).	<pre>n = int(input("Enter n > 1: ")) n_divisors = 0 for i in range(1, n + 1): if n % i == 0: # n chia hết cho i n_divisors += 1 if n_divisors == 2: print("n is prime") else: print("n is composite")</pre>

Số nguyên tố là gì?

Tiếng Việt	Python
Cho số nguyên $n > 1$, n được gọi là số nguyên tố (prime) nếu n không chia hết cho số nguyên nào trong khoảng $[2, n - 1]$, ngược lại n được gọi là hợp số (composite).	<pre>n = int(input("Enter n > 1: ")) is_prime = True for i in range(2, n): if n % i == 0: # n chia hết cho i is_prime = False break if is_prime: print("n is prime") else: print("n is composite")</pre>

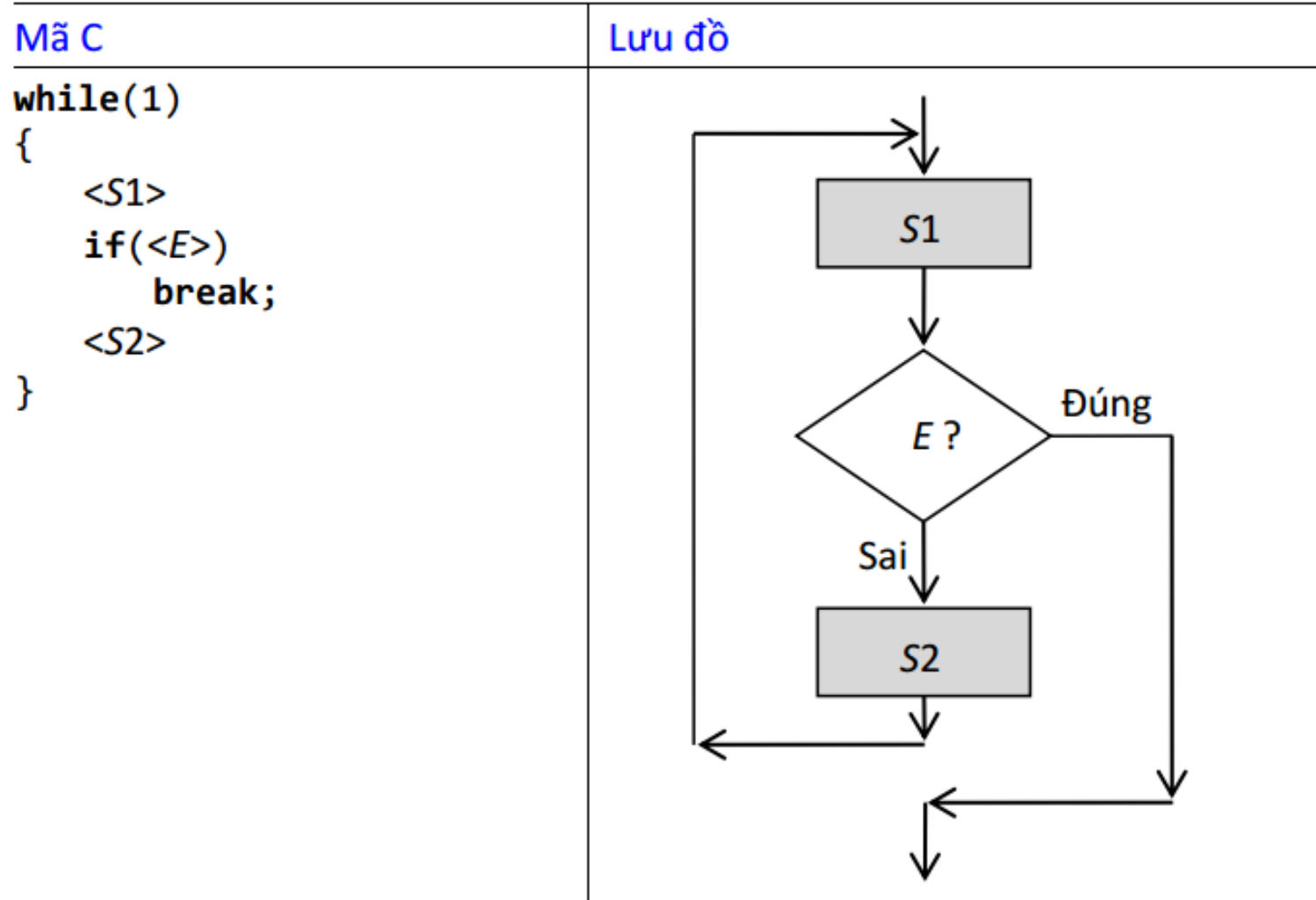
Số nguyên tố là gì?

C	C++
<pre>#include <stdio.h> int main() { int n; printf("Enter n > 1: "); scanf("%d", &n); int is_prime = 1; // true for (int i = 2; i < n; i++) if (n % i == 0) { // n chia het cho i is_prime = 0; // false break; } if (is_prime) printf("n is prime"); else printf("n is composite"); }</pre>	<pre>#include <iostream> using namespace std; int main() { int n; cout << "Enter n > 1: "; cin >> n; bool is_prime = true; for (int i = 2; i < n; i++) if (n % i == 0) { // n chia het cho i is_prime = false; break; } if (is_prime) cout << "n is prime"; else cout << "n is composite"; }</pre>

Có thể điều khiển chi tiết các vòng lặp không?

- Có 2 lệnh điều khiển thường được dùng để điều khiển chi tiết các vòng lặp là:
 - **break** giúp bỏ qua các lệnh còn lại trong thân lặp và thoát khỏi vòng lặp
 - **continue** giúp bỏ qua các lệnh còn lại trong thân lặp và chuyển qua lần lặp kế tiếp

Lệnh break khác lệnh continue thế nào?



Lệnh break khác lệnh continue thế nào?

Mã C	Lưu đồ
<pre>while(1) { <S1> if(<E>) continue; <S2> }</pre>	<pre>graph TD Entry(()) --> S1[S1] S1 --> E{E ?} E -- Đúng --> Exit(()) E -- Sai --> S2[S2] S2 --> Entry</pre> <p>The flowchart illustrates the execution of the provided C code. It starts with an entry point leading to a process box labeled S1. Following S1, there is a decision diamond labeled E?. If the condition E is true (Đúng), the flow proceeds to an exit point. If the condition E is false (Sai), the flow proceeds to a process box labeled S2. After S2, the flow loops back to the entry point before S1, forming a continuous cycle.</p>

Vòng lặp “giả vô hạn” là gì?

- Lệnh break cũng thường được dùng trong các cấu trúc lặp “giả vô hạn” (pseudo-infinite loop)

```
while True:
```

```
    <S>
```

```
    if <C>:
```

```
        break
```

```
while (1){
```

```
    <S>
```

```
    if (<C>)
```

```
        break;
```

```
}
```

- Chiến lược **đánh giá/lượng giá muộn** (lazy/deferred evaluation):
không xác định điều kiện lặp từ đầu mà xác định khi cần

Vẽ rùa (turtle drawing) là gì?

- Module turtle của Python hỗ trợ **đồ họa** (graphic) đơn giản
- <https://docs.python.org/3/library/turtle.html>
- *Vẽ rùa là một công cụ học lập trình rất bổ ích!*

Cộng điểm

- Làm bài tập 7.7.(a) (Tài liệu Python)