# VÒNG LẶP

## VÒNG LẶP FOR

### Khái niệm

Mục đích của vòng lặp là để giải quyết các hành động (khối code) có sự lặp đi lặp lại nhiều lần với số lần đã biết trước.

### Cú pháp

|  |
| --- |
| for (Khởi tạo giá trị biến lặp; Điều kiện lặp; Cập nhật biến lặp)  {  *// Khối lệnh lặp*  } |

Trong đó:

Vòng lặp for được định nghĩa bởi từ khóa for và được chia làm 3 phần chính, mỗi phần được ngăn cách bởi dấu chấm phẩy(;)

* **Phần khởi tạo biến lặp**: Kởi tạo biến vòng lặp có thể được khai báo và khởi tạo giá trị ngay bên trong phần khởi tạo của vòng lặp for. Phần khởi tạo biến được thực thi đầu tiên và chỉ thực thi 1 lần duy nhất trong vòng lặp for.
* **Điều kiện lặp:** Khối lệnh của vòng lặp for sẽ được thực hiện nếu **Biểu thức điều kiện lặp** cho giá trị đúng. Vòng lặp for kiểm tra biểu thức điều kiện trước khi thực hiện khối lệnh.
* **Cập nhật biến lặp:** Phần này sẽ được thực thi cuối mỗi lần lặp, sau khi khối lệnh của vòng lặp for được thực thi. Phần này thường chịu trách nhiệm thay đổi giá trị biến vòng lặp được sử dụng trong biểu thức điều kiện (nhằm tránh tình trạng lặp vô hạn). Sau khi thực thi xong phần cập nhật biến vòng lặp, chương trình quay trở lại đánh giá biểu thức điều kiện của vòng lặp **for** và cứ như thế đến khi biểu thức điều kiện lặp là sai.

**Chú ý:**

* Có dấu chấm phẩy sau lệnh for (i = 0; i <3; i++); → vòng lặp for sẽ không chạy đúng chương trình.
* Trong vòng lặp for có sử dụng từ 2 lệnh trở lên, nhớ sử dụng cặp ngoặc { } để bọc các lệnh đó lại.

**Lưu đồ:**

A picture containing text, diagram, line, plan

Description automatically generated

**Bước 1**: Khởi tạo giá trị biến lặp (1 lần duy nhất)

**Bước 2**: Kiểm tra điều kiện lặp, nếu sai chuyển sang Bước 5, nếu đúng chuyển sang Bước 3.

**Bước 3**: Thực hiện nội dung trong thân vòng lặp

**Bước 4**: Cập nhật giá trị lặp và quay lại Bước 2

**Bước 5**: Kết thúc vòng lặp.

|  |
| --- |
| **VD 1**: Viết 1 chương trình in ra câu "Toi se hoc gioi C " 3 lần. |
| #include<stdio.h>  int main(){      for (int i = 0; i < 3; i++) {          printf("Toi se hoc gioi C\n");      }      return 0;  } |
| **Kết quả:** |
| Toi se hoc gioi C  Toi se hoc gioi C  Toi se hoc gioi C |

|  |
| --- |
| **VD 2:** Viết chương trình nhập vào n số nguyên và tính tổng từ 1 n. |
| #include<stdio.h>  int main (){      int n, tong = 0;      printf("Nhap vao so nguyen: ");      scanf("%d", &n);        for (int i = 1; i <= n; i++) {          tong += i;      }      printf ("Tong la: %d",tong);      return 0;  } |
| **Kết quả:** |
| Nhap vao so nguyen: 5  Tong la: 15 |

|  |
| --- |
| **VD 3:** Viết chương trình nhập vào số nguyên n. Tính tổng các giá trị lẻ từ 0 đến n. |
| #include<stdio.h>  int main (){      int n, tong = 0;      printf("Nhap vao so nguyen: ");      scanf("%d", &n);        for (int i = 1; i <= n; i++)          if(i %2 == 1)              tong += i;      printf ("Tong la: %d",tong);      return 0;  } |
| **Kết quả:** |
| Nhap vao so nguyen: 5  Tong la: 9 |

### Lược bỏ 1 số thành phần trong vòng lặp for

#### Lược bỏ biểu thức khởi tạo giá trị biến lặp

Nếu bỏ qua ***biểu thức khởi tạo***, chương trình sẽ sử dụng giá trị của biến được khai báo trước đó hoặc giá trị mặc định của kiểu dữ liệu.

|  |
| --- |
| **VD 5:** Tính tổng các số từ 1 tới n |
| #include <stdio.h>  int main() {      int i = 1, n, tong = 0; // i là biến được khai báo trước vòng lặp for      printf("Nhap n: ");      scanf("%d", &n);      for ( ; i <= n; i++) {          tong += i;      }      printf("Tong tu 1 den %d la: %d", n, tong);      return 0;  } |
| **Kết quả:** |
| Nhap n: 5  Tong tu 1 den 5 la: 15 |

#### Lược bỏ biểu thức điều kiện lặp

Nếu bỏ qua ***biểu thức điều kiện lặp***, vòng lặp sẽ lặp vô tận nếu không có câu lệnh break trong khối lệnh.

|  |
| --- |
| **VD 6:** In ra các số từ 0 tới 5 |
| #include <stdio.h>  int main() {      int i;      for (i = 0; ; i++) {          if (i > 5) {              break;          }          printf("Gia tri: %d\n", i);      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 1  Gia tri: 2  Gia tri: 3  Gia tri: 4  Gia tri: 5 |

#### Lược bỏ biểu thức cập nhật biến lặp

Nếu thiếu ***biểu thức cập nhật biến lặp***, chương trình sẽ luôn luôn chạy cho đến khi tràn bộ nhớ. Vì vậy ta sẽ cập nhật biến lặp bên trong khối lệnh của vòng lặp.

|  |
| --- |
| **VD7:** Tính tổng các số từ 1 tới n |
| #include <stdio.h>  int main() {      int n, tong = 0;      printf("Nhap n: ");      scanf("%d", &n);      for (int i = 1; i <= n; ) {          tong += i;          i++;      }      printf("Tong tu 1 den %d la: %d", n, tong);      return 0;  } |
| **Kết quả:** |
| Nhap n: 5  Tong tu 1 den 5 la: 15 |

#### Lược bỏ biểu thức khởi tạo giá trị biến lặp và biểu thức cập nhật biến lặp

Nếu vắng cả biểu thức khởi tạo và biểu thức tăng/giảm, ta có thể dùng một biến trước đó để khởi tạo giá trị ban đầu và sử dụng biến đó trong vòng lặp, sau đó tăng/giảm giá trị của biến trong khối lệnh của vòng lặp

|  |
| --- |
| **VD 8:** Tính tổng các số từ 1 tới n |
| #include <stdio.h>  int main() {      int n;      printf("Nhap n: ");      scanf("%d", &n);      int i = 1;      int tong = 0;      for ( ; i <= n; ) {          tong += i;          i++;      }      printf("Tong tu 1 den %d la: %d", n, tong);      return 0;  } |
| **Kết quả:** |
| Nhap n: 5  Tong tu 1 den 5 la: 15 |

#### Lược bỏ biểu thức điều kiện lặp và biểu thức khởi tạo giá trị biến lặp

Nếu lược bỏ cả biểu thức điều kiện lặp và biểu thức khởi tạo giá trị biến lặp trong vòng lặp for, ta có thể dùng một biến trước đó để khởi tạo giá trị ban đầu và sử dụng biến đó trong vòng lặp và ta cần sử dụng một vòng lặp vô hạn và cần phải sử dụng lệnh break để thoát khỏi vòng lặp.

|  |
| --- |
| **VD 9:** In ra các số từ 0 đến 5 |
| #include <stdio.h>  int main() {      int i = 0;      for ( ; ; i++) {          if (i > 5) {              break;          }          printf("Gia tri: %d\n", i);      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 1  Gia tri: 2  Gia tri: 3  Gia tri: 4  Gia tri: 5 |

#### Lược bỏ biểu thức điều kiện lặp và biểu thức cập nhật biến lặp

Nếu lược bỏ cả biểu thức điều kiện lặp và biểu thức cập nhật biến lặp trong vòng lặp for, thì ta cần sử dụng một vòng lặp vô hạn và cần phải sử dụng lệnh break để thoát khỏi vòng lặp. tăng/giảm giá trị của biến trong khối lệnh của vòng lặp

|  |
| --- |
| **VD 9:** In ra các số từ 0 đến 5 |
| #include <stdio.h>  int main() {      for (int i = 0; ; ) {          if (i > 5) {              break;          }          printf("Gia tri: %d\n", i);          i++;      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 1  Gia tri: 2  Gia tri: 3  Gia tri: 4  Gia tri: 5 |

#### Vòng lặp for vắng mặt cả 3 biểu thức

Nếu vòng lặp for vắng mặt cả 3 biểu thức (khởi tạo, điều kiện lặp và cập nhật) thì sẽ tạo ra một vòng lặp vô hạn và lệnh trong khối lệnh sẽ được thực thi vô tận cho đến khi có một lệnh break hoặc return được sử dụng để thoát khỏi vòng lặp.

|  |
| --- |
| **VD 11:** In ra các số từ 0 đến 5 |
| #include <stdio.h>  int main() {      int i = 0;      for ( ; ; ) {          if (i > 5) {              break;          }          printf("Gia tri: %d\n", i);          i++;      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 1  Gia tri: 2  Gia tri: 3  Gia tri: 4  Gia tri: 5 |

## VÒNG LẶP WHILE

### Khái niệm

Được sử dụng để lặp đi lặp lại một hành động (khối lệnh) mà không biết trước số lần lặp.

### Cú pháp

|  |
| --- |
| while (Biểu thức) {  // Khối lệnh lặp  } |

Trong đó:

* Từ khóa while phải viết bằng chữ thường
* Nếu khối lệnh bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

**Lưu đồ:**

**A diagram of a flowchart

Description automatically generated with low confidence**

Kiểm tra biểu thức:

* Nếu **sai** thì kết thúc vòng lặp while (khối lệnh không được thi hành 1 lần nào)
* Nếu **đúng** thì thực hiện khối lệnh Lặp lại ***Kiểm tra biểu thức***.

**Chú ý:**

* Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.
* Trong thân while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.
* Trong thân while có thể sử dụng lệnh continue để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).
* Muốn thoát khỏi vòng lặp while tùy ý có thể dùng các lệnh break, goto, return như lệnh for.

|  |
| --- |
| **VD 1:** Viết 1 chương trình in ra các số từ 1 n, n được nhập từ bàn phím |
| #include <stdio.h>  int main(){      int n, i = 0;      printf ("Nhap n: ");      scanf ("%d", &n);      while (i < n) {          i++;          printf("Gia tri i: %d\n",i);      }      return 0;  } |
| **Kết quả:** |
| Nhap n: 5  Gia tri i: 1  Gia tri i: 2  Gia tri i: 3  Gia tri i: 4  Gia tri i: 5 |

|  |
| --- |
| **VD 2:** Chương trình tính tổng các số nguyên từ 1 n |
| #include <stdio.h>  int main() {      int i = 0, n, tong = 0;      printf("Nhap vao so n: ");      scanf("%d", &n);      while (i++ < n) {          tong = tong + i; *//hoac tong += i;*      }      printf("Tong: %d", tong);      return 0;  } |
| **Kết quả:** |
| Nhap vao so n: 5  Tong: 15 |
| **Lưu ý:** |
| Trong biểu thức (i++ < n). Chương trình sẽ kiểm tra kiểm thức <n trước rồi mới cập nhật giá trị i++ |

|  |
| --- |
| **VD 3:** Viết 1 chương trình in ra chuỗi số giảm dần từ 6 về 0 |
| #include <stdio.h>  int main(){      int n = 5, tong = 0;      while (n--) { *// n = 0*          tong = tong + n; *// hoac tong += i;   tong = 10*          printf("Gia tri i: %d\n", n);      }      printf("Tong : %d\n", tong);      return 0;  } |
| **Kết quả:** |
| 5  4  3  2  1  0 |
| **Lưu ý:** |
| Trong biểu thức (n--). Chương trình sẽ kiểm tra kiểm thức n trước rồi mới cập nhật giá trị n-- |

|  |
| --- |
| **VD 4:** In ra tất cả số chẵn từ 1 đến 10. |
| #include <stdio.h>  int main() {      int i = 0;      while (i <= 10){          printf("Gia tri: %d\n", i);          i+=2;      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 2  Gia tri: 4  Gia tri: 6  Gia tri: 8  Gia tri: 10 |

## VÒNG LẶP DO WHILE

### Khái niệm

Vòng lặp do…while là vòng lặp mà chương trình sẽ thực thi ít nhất một lần.

### Cú pháp

|  |
| --- |
| do {  // Khối lệnh  }  while (Biểu thức); |

Trong đó:

* Từ khóa do, while phải viết bằng chữ thường
* Nếu khối lệnh bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }

**Lưu đồ**

**A picture containing diagram, line, text, font

Description automatically generated**

**Bước 1**: Thực hiện khối lệnh.

**Bước 2**: Kiểm tra biểu thức:

* Nếu biểu thức là **Đúng** thì quay lại Bước 1.
* Nếu biểu thức là **Sai** thì chuyển sang Bước 3 (khối lệnh được thi hành 1 lần).

**Bước 3**: Kết thúc vòng lặp

**Chú ý:**

* Biểu thức: có thể là một biểu thức hoặc nhiều biểu thức con. Nếu là nhiều biểu thức con thì cách nhau bởi dấu phẩy (,) và tính đúng sai của biểu thức được quyết định bởi biểu thức con cuối cùng.
* Trong thân do…while (khối lệnh) có thể chứa một hoặc nhiều cấu trúc điều khiển khác.
* Trong thân do…while có thể sử dụng lệnh **continue** để chuyển đến đầu vòng lặp (bỏ qua các câu lệnh còn lại trong thân).
* Muốn thoát khỏi vòng lặp do…while tùy ý có thể dùng các lệnh **break, goto, return.**

|  |
| --- |
| **VD 1:** Viết chương trình kiểm tra passwork |
| #include <stdio.h>  # define PASSWORD 12345  int main() {      int n;      do {          printf("Nhap vao password: ");          scanf("%d", &n);      }      while (n != PASSWORD);      printf("Xong !");      return 0;  } |
| **Kết quả:** |
| Nhap vao password: 123  Nhap vao password: 12345  Xong ! |

|  |
| --- |
| **VD 2:** Viết chương trình nhập vào số nguyên dương, nếu không phải thì nhập lại. |
| #include <stdio.h>  int main() {      int n;      do {          printf("Nhap vao so nguyen: ");          scanf("%d", &n);      }      while (n < 0);      printf("Xong !");      return 0;  } |
| **Kết quả:** |
| Nhap vao so nguyen: -99  Nhap vao so nguyen: 10  Xong ! |

|  |
| --- |
| **VD 3:** In ra tất cả số chẵn từ 1 đến 10. |
| #include <stdio.h>  int main() {      int i;      while (i <= 10){          printf("Gia tri: %d\n", i);          i+=2;      }      return 0;  } |
| **Kết quả:** |
| Gia tri: 0  Gia tri: 2  Gia tri: 4  Gia tri: 6  Gia tri: 8  Gia tri: 10 |

## VÒNG LẶP LỒNG NHAU

Ta cũng có thể đặt một vòng lặp bên trong một vòng lặp khác. Đây được gọi là vòng lặp lồng nhau. "Vòng lặp bên trong" sẽ được thực hiện một lần cho mỗi lần lặp lại "vòng lặp bên ngoài”

|  |
| --- |
| **VD 1:** Vẽ hình chữ nhật đặc bằng các dấu '\*' |
| #include<stdio.h>  int main(){      int a,b;      printf("Nhap vao chieu dai HCN: ");      scanf("%d",&a);      printf("Nhap vao chieu rong HCN: ");      scanf("%d",&b);      for (int i = 1; i <= b; i++){          for (int j = 1; j <= a; j++){              printf("\*");          }          printf("\n");      }      return 0;  } |
| **Kết quả:** |
| Nhap vao chieu dai HCN: 9  Nhap vao chieu rong HCN: 5  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\* |

Các lệnh lặp for, while, do…while có thể lồng vào chính nó, hoặc lồng vào lẫn nhau. Nếu không cần thiết không nên lồng vào nhiều cấp dễ gây nhầm lẫn khi lập trình cũng như kiểm soát chương trình.

***So sánh sự khác nhau của các vòng lặp***

* + Vòng lặp for thường sử dụng khi biết được số lần lặp xác định.
  + Vòng lặp thường while, do…while sử dụng khi không biết rõ số lần lặp.
  + Khi gọi vòng lặp while, do…while, nếu biểu thức sai vòng lặp while sẽ không được thực hiện lần nào nhưng vòng lặp **do…while** thực hiện được 1 lần. )

**Số lần thực hiện ít nhất của while là 0 và của do…while là 1**

## CÂU LỆNH ĐIỀU KHIỂN

### Lệnh break

Thông thường lệnh break dùng để thoát khỏi vòng lặp không xác định điều kiện dừng hoặc bạn muốn dừng vòng lặp theo điều kiện do bạn chỉ định. Việc dùng lệnh break để thoát khỏi vòng lặp thường sử dụng phối hợp với lệnh if. Lệnh break dùng trong for, while, do…while, switch. Lệnh break thoát khỏi vòng lặp chứa nó.

|  |
| --- |
| **VD 1:** In những số < 10 trong khoảng [0;20] |
| #include <stdio.h>  int main(){      int i;      for (i = 0; i < 20; i++){          if(i>=10)              break;          printf("%d ",i);      }      return 0;  } |
| **Kết quả:** |
| 0 1 2 3 4 5 6 7 8 9 |

### Lệnh continue

Lệnh continue được dùng trong vòng lặp for, while, do…while. Khi lệnh continue thi hành quyền điều khiển sẽ trao qua cho biểu thức điều kiện của vòng lặp gần nhất. Nghĩa là lộn ngược lên đầu vòng lặp, tất cả những lệnh đi sau trong vòng lặp chứa continue sẽ bị bỏ qua không thi hành.

|  |
| --- |
| **VD 2:** In ra các chữ số lẻ từ 1 đến 10 |
| #include <stdio.h>  int main(){      for(int i = 0; i <= 10; i++){          if(i % 2 == 0){              continue;          }          printf("Gia tri: %d\n", i);      }  } |
| **Kết quả:** |
| 1 3 5 7 9 |

### Lệnh goto

Câu lệnh goto trong C cung cấp một bước nhảy vô điều kiện từ 'goto' đến một câu lệnh có nhãn là ‘label’.

**Chú ý**:

Việc sử dụng câu lệnh goto không được khuyến khích sử dụng trong bất kỳ ngôn ngữ lập trình nào vì nó rất khó để theo dõi luồng điều khiển của chương trình, làm cho chương trình khó hiểu và khó bảo trì.

|  |
| --- |
| **VD 3:** |
|  |
| **Kết quả:** |
|  |