



CẤU TRÚC RẼ NHÁNH

IF...ELSE

SWITCH CASE

NỘI DUNG BÀI HỌC

01. Câu lệnh if

02. Câu lệnh if else

03. Câu lệnh if và else if

04. Switch case

05. Bảng mã ASCII và các câu lệnh kiểm tra loại ký tự





Hướng dẫn

Các bạn đọc slide này kèm với thực hành bài tập phần Cấu trúc rẽ nhánh (45 bài) trên website OJ tại địa chỉ :

http://oj.28tech.com.vn/problems/?category=3&point_start=&point_end=&order=code



1. Câu lệnh if

- ▶▶▶ Câu lệnh if được sử dụng trong trường hợp bạn muốn chương trình của mình thực hiện 1 hoặc 1 nhóm câu lệnh khi một điều kiện nào đó thỏa mãn.
- ▶▶▶ VD: Nếu chỉ số máu của nhân vật bằng 0 thì nhân vật sẽ chết, vậy điều kiện ở đây là “chỉ số máu của nhân vật bằng 0”, và hành động được thực hiện ở đây là sẽ có một câu lệnh thực thi làm nhân vật bị chết.

SYNTAX

```
if (condition) {  
    // code  
}
```

1. Câu lệnh if

Giải thích : Điều kiện trong **if** thứ nhất là **true** vì thế 2 câu lệnh bên trong **if** đó sẽ được thực hiện. Điều kiện trong **if** thứ 2 có giá trị **false** vì thế 2 câu lệnh in ra python và html sẽ không được thực hiện

Ví dụ 1

```
#include <iostream>

using namespace std;

int main(){
    cout << "Truoc\n";
    if(true){
        cout << "28tech.com.vn\n";
        cout << "lap trinh\n";
    }
    if(false){
        cout << "python\n";
        cout << "html\n";
    }
    cout << "Sau\n";
    return 0;
}
```

OUTPUT

```
Truoc
28tech.com.vn
lap trinh
Sau
```



1. Câu lệnh if

Giải thích : Điều kiện trong **if** thứ nhất là **0** vì thế 2 câu lệnh bên trong **if** đó sẽ không được thực hiện. Điều kiện trong **if** thứ 2 có giá trị **5** vì thế 2 câu lệnh in ra python và html sẽ được thực hiện.

Chú ý : Trong C++ thì giá trị khác 0 sẽ được coi là đúng (true) còn giá trị 0 sẽ được coi là false

Ví dụ 2

```
#include <iostream>

using namespace std;

int main(){
    cout << "Truoc\n";
    if(0){
        cout << "28tech.com.vn\n";
        cout << "lap trinh\n";
    }
    if(5){
        cout << "python\n";
        cout << "html\n";
    }
    cout << "Sau\n";
    return 0;
}
```

OUTPUT

```
Truoc
python
html
Sau
```



1. Câu lệnh if

Giải thích : Điều kiện trong **if** thứ nhất là $n > 3$ mà $n = 15$ nên phép so sánh này có giá trị đúng, 2 câu lệnh bên trong **if** này được thực hiện. Điều kiện trong **if** thứ 2 tương đương với $\text{true} \ \&\& \ \text{false} = \text{false}$ nên 2 câu lệnh in python và html không được thực hiện.

Ví dụ 3

```
#include <iostream>

using namespace std;

int main(){
    int n = 15;
    if(n > 3){
        cout << "28tech.com.vn\n";
        cout << "lap trinh\n";
    }
    if((n > 3) && (n < 10)){
        cout << "python\n";
        cout << "html\n";
    }
    return 0;
}
```

OUTPUT

```
28tech.com.vn
lap trinh
```

1. Câu lệnh if

MỘT SỐ CÂU LỆNH IF THƯỜNG GẶP

Điều kiện kiểm tra	Câu lệnh
1. Kiểm tra n là số chẵn	<code>if ((n % 2) == 0)</code>
2. Kiểm tra n là số lẻ	<code>if ((n % 2) == 1), if((n % 2) != 0)</code>
3. Kiểm tra n chia hết cho k	<code>if ((n % k) == 0)</code>
4. Kiểm tra n không chia hết cho k	<code>if ((n % k) != 0)</code>
5. Kiểm tra n chia hết cho cả a và b	<code>if ((n % a) == 0 && ((n % b) == 0))</code>
6. Kiểm tra n chia hết cho a hoặc b	<code>if ((n % a) == 0 ((n % b) == 0))</code>
7. Kiểm tra n thuộc đoạn [a, b]	<code>if ((n >= a) && (n <= b))</code>



1. Câu lệnh if MỘT SỐ CÂU LỆNH IF THƯỜNG GẶP

Điều kiện kiểm tra	Câu lệnh
8. Kiểm tra n là số nguyên tố có 1 chữ số	<code>if ((n == 2) (n == 3) (n == 5) (n == 7))</code>
9. Kiểm tra n là số có k chữ số	<code>if ((n >= pow(10, k - 1) && (n < pow(10, k)))</code>
10. Kiểm tra n là số khác 0	<code>if (n != 0) hoặc if(n)</code>
11. Kiểm tra biến bool check có giá trị true	<code>if (check == true) hoặc if(check)</code>
12. Kiểm tra biến bool check có giá trị false	<code>if (check == false) hoặc if(!check)</code>



Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse06>



2. Câu lệnh if else

- If được sử dụng khi bạn muốn thực thi code với điều kiện nào đó đúng, trong trường hợp điều kiện đó sai bạn muốn thực thi một đoạn code khác thì cấu trúc if else sẽ được sử dụng



SYNTAX

```
if (condition) {  
    // code if  
}  
else {  
    // code else  
}
```

2. Câu lệnh if else

Giải thích : Điều kiện trong if tương đương với $15 \% 2 == 1$ sẽ có giá trị true vì thế 2 câu lệnh trong if được thực thi. Cấu trúc if else chỉ thực hiện 1 trong 2 nhánh.

Ví dụ 1

```
#include <iostream>

using namespace std;

int main(){
    int n = 15;
    if((n % 2) == 1){
        cout << "28tech.com.vn\n";
        cout << "lap trinh\n";
    }
    else{
        cout << "python\n";
        cout << "html\n";
    }
    return 0;
}
```

OUTPUT

```
28tech.com.vn
lap trinh
```

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse09>





3. Câu lệnh if và else if

- ▶▶▶ Nếu bạn muốn kiểm tra nhiều điều kiện khác nhau thì sử dụng cấu trúc else if sẽ hiệu quả hơn so với sử dụng nhiều câu lệnh if else lồng nhau.



SYNTAX

```
if (condition1) {  
    // code  
}  
else if (condition2){  
    // code  
}  
....  
else if(conditionN){  
    //code  
}  
else{  
    // code  
}
```

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse14>



3. Câu lệnh if và else if

Chú ý : Nếu một điều kiện nào trong N điều kiện trong cấu trúc trên đúng và câu lệnh bên trong nhánh đó được thực hiện thì khối lệnh if else if này sẽ kết thúc ngay lập tức. Ví dụ, nếu condition2 đúng thì khối lệnh bên trong nhánh đó được thực thi, sau đó cấu trúc này sẽ kết thúc ngay mà không kiểm tra các điều kiện còn lại cũng như trong else.

Good	Bad
<pre>int day; cin >> day; if(day == 1) cout << "Chu nhat" << endl; else if(day == 2) cout << "Thu hai" << endl; else if(day == 3) cout << "Thu ba" << endl; else if(day == 4) cout << "Thu tu" << endl; else if(day == 5) cout << "Thu nam" << endl; else if(day == 6) cout << "Thu sau" << endl; else if(day == 7) cout << "Thu bay" << endl;</pre>	<pre>int day; cin >> day; if(day == 1) cout << "Chu nhat" << endl; if(day == 2) cout << "Thu hai" << endl; if(day == 3) cout << "Thu ba" << endl; if(day == 4) cout << "Thu tu" << endl; if(day == 5) cout << "Thu nam" << endl; if(day == 6) cout << "Thu sau" << endl; if(day == 7) cout << "Thu bay" << endl;</pre>

4. Switch case

- ▶▶▶ **Ý nghĩa** : Giá trị của val sẽ được so sánh lần lượt với các giá trị trong các case, nếu giá trị của val bằng giá trị tại 1 case nào đó thì câu lệnh bên trong case đó được thực thi. Nếu val không giống bất cứ một giá trị trong các case nào thì câu lệnh bên trong default được thực thi.
- ▶▶▶ **Chú ý** : Giá trị của val có thể là số, kí tự, xâu kí tự (sẽ học sau) .Các khối lệnh bên trong các case sẽ được kết thúc bằng câu lệnh break.

SYNTAX

```
switch (val) {  
    case 1:  
        // code  
        break;  
    case 2:  
        //code  
        break;  
    .....  
    case n:  
        // code  
        break;  
    default :  
        // code  
}
```

4. Switch case

Giải thích : Giá trị trong **switch** là n, khi đó giá trị của n bằng với giá trị của case 3, do đó câu lệnh trong **case 3** được thực thi. Tuy nhiên **switch case** khi đã thực hiện câu lệnh trong nhánh nào sẽ thực hiện tiếp các nhánh bên dưới của nó. Vì thế ở mỗi **case** cần có câu lệnh **break** ở cuối cùng.

Ví dụ 1

```
#include <iostream>
using namespace std;

using ll = long long;

int main(){
    int n = 3;
    switch(n){
        case 1:
            cout << "python\n";
        case 2:
            cout << "php\n";
        case 3:
            cout << "javascript\n";
        case 4:
            cout << "go\n";
        default:
            cout << "html\n";
    }
}
```

OUTPUT

```
javascript
go
html
```



4. Switch case

Giải thích : Giá trị trong **switch** là kí tự c, khi đó giá trị của c bằng với giá trị trong **case** thứ 1, khi đó câu lệnh trong **case** thứ nhất được in ra. Trong trường hợp này thì **switch case** đã có câu lệnh **break** ở cuối vì thế nó sẽ không thực hiện các câu lệnh ở trong các nhánh bên dưới.

Ví dụ 2

```
#include <iostream>
using namespace std;

using ll = long long;

int main(){
    int a = 10, b = 5;
    char c = '+';
    switch(c){
        case '+':
            cout << a + b << endl;
            break;
        case '-':
            cout << a - b << endl;
            break;
        case '*':
            cout << a * b << endl;
            break;
        case '/':
            cout << a / b << endl;
            break;
    }
}
```

OUTPUT

15

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse43>

4. Switch case

Giải thích : Trong trường hợp có nhiều case có cùng các câu lệnh cần thực thi, bạn có thể gom chung nó lại làm 1. Ví dụ trên là in ra số lượng ngày trong 1 tháng bỏ qua trường hợp năm nhuận. Bạn làm bài tập kiểm tra số ngày của tháng bằng switch case để luyện tập thêm.

Ví dụ 3

```
#include <iostream>
using namespace std;
```

```
using ll = long long;
```

```
int main(){
    int m = 6;
    switch(m){
        case 1 : case 3 : case 5 : case 7 :
        case 8 : case 10 : case 12:
            cout << 31 << endl;
            break;
        case 4 : case 6 : case 9 : case 11:
            cout << 30 << endl;
            break;
        case 2:
            cout << 28 << endl;
            break;
        default:
            cout << "invalid\n";
    }
}
```

OUTPUT

30

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse12>

4. Switch case

Giải thích : Trong trường hợp cần check giá trị trong **switch** là số nguyên trong đoạn từ **x** tới **y** bạn có thể sử dụng cú pháp **x ... y**. Trong ví dụ khi **x = 15** thuộc đoạn từ 11 tới 20 nên nhánh thứ 2 sẽ được thực hiện.

Ví dụ 4

```
#include <iostream>
using namespace std;
```

```
using ll = long long;
```

```
int main(){
    int x = 15;
    switch (x){
        case 1 ... 10:
            cout << "28tech.com.vn\n";
            break;
        case 11 ... 20:
            cout << "blog.28tech.com.vn\n";
            break;
        case 21 ... 40:
            cout << "khoahoc.28tech.com.vn\n";
            break;
        default:
            cout << "facebook.com\n";
    }
}
```

OUTPUT

blog.28tech.com.vn

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse42>



28TECH

Become A Better Developer

BẢNG MÃ ASCII VÀ CÁC LỆNH LIÊN QUAN



5. Bảng mã ASCII và các câu lệnh kiểm tra loại ký tự

0		20	␣	40	<	60	<	80	P	100	d	120	x	140	î	160	à	180	␣	200	␣	220	␣	240	␣
1	␣	21	␣	41	>	61	=	81	Q	101	e	121	y	141	ï	161	á	181	␣	201	␣	221	␣	241	␣
2	␣	22	␣	42	*	62	>	82	R	102	f	122	z	142	ä	162	â	182	␣	202	␣	222	␣	242	␣
3	␣	23	␣	43	+	63	?	83	S	103	g	123	<	143	å	163	ä	183	␣	203	␣	223	␣	243	␣
4	␣	24	␣	44	,	64	@	84	T	104	h	124	>	144	æ	164	å	184	␣	204	␣	224	␣	244	␣
5	␣	25	␣	45	-	65	A	85	U	105	i	125	Δ	145	␣	165	␣	185	␣	205	␣	225	␣	245	␣
6	␣	26	␣	46	.	66	B	86	V	106	j	126	␣	146	␣	166	␣	186	␣	206	␣	226	␣	246	␣
7	␣	27	␣	47	/	67	C	87	W	107	k	127	␣	147	␣	167	␣	187	␣	207	␣	227	␣	247	␣
8	␣	28	␣	48	0	68	D	88	X	108	l	128	␣	148	␣	168	␣	188	␣	208	␣	228	␣	248	␣
9	␣	29	␣	49	1	69	E	89	Y	109	m	129	␣	149	␣	169	␣	189	␣	209	␣	229	␣	249	␣
10	␣	30	␣	50	2	70	F	90	Z	110	n	130	␣	150	␣	170	␣	190	␣	210	␣	230	␣	250	␣
11	␣	31	␣	51	3	71	G	91	[111	o	131	␣	151	␣	171	␣	191	␣	211	␣	231	␣	251	␣
12	␣	32	␣	52	4	72	H	92	\	112	p	132	␣	152	␣	172	␣	192	␣	212	␣	232	␣	252	␣
13	␣	33	␣	53	5	73	I	93]	113	q	133	␣	153	␣	173	␣	193	␣	213	␣	233	␣	253	␣
14	␣	34	␣	54	6	74	J	94	^	114	r	134	␣	154	␣	174	␣	194	␣	214	␣	234	␣	254	␣
15	␣	35	␣	55	7	75	K	95	_	115	s	135	␣	155	␣	175	␣	195	␣	215	␣	235	␣	255	␣
16	␣	36	␣	56	8	76	L	96	`	116	t	136	␣	156	␣	176	␣	196	␣	216	␣	236	␣		
17	␣	37	␣	57	9	77	M	97	a	117	u	137	␣	157	␣	177	␣	197	␣	217	␣	237	␣		
18	␣	38	␣	58	:	78	N	98	b	118	v	138	␣	158	␣	178	␣	198	␣	218	␣	238	␣		
19	␣	39	␣	59	;	79	O	99	c	119	w	139	␣	159	␣	179	␣	199	␣	219	␣	239	␣		



5. Bảng mã ASCII và các câu lệnh kiểm tra loại ký tự

Bảng mã này có 256 ký tự, mỗi ký tự được gán với 1 mã nhất định gọi là mã ASCII. Bạn có thể coi kiểu dữ liệu char như số hoặc ký tự đều được, tức là bạn hoàn toàn có thể sử dụng nó để cộng, trừ, nhân, chia

Chú ý: Hãy luôn nhớ khi cộng, trừ, nhân, chia một ký tự nào đó thì mã ASCII của nó sẽ được sử dụng.

Một vài dải ký tự cần lưu ý:

Dải ký tự	Dải mã ASCII
A-Z	65-90
a-z	97-122
0-9	48-57

Bài tập áp dụng : <http://oj.28tech.com.vn/problem/ifelse17>

5. Bảng mã ASCII và các câu lệnh kiểm tra loại ký tự

Giải thích : Khi in ra giá trị của c dưới dạng số int thì nó sẽ ép sang mã ASCII của ký tự đó. Ký tự 'A' trong bảng mã có mã là 65. Bạn cũng có thể sử dụng biến char trong các phép tính toán, khi đó mã ASCII sẽ được sử dụng, từ đó x sẽ có giá trị 75.

Ví dụ 1

```
#include <iostream>
using namespace std;

using ll = long long;

int main(){
    char c = 'A';
    cout << c << endl;
    cout << (int) c << endl;
    int x = c + 10;
    cout << x << endl;
    ++c;
    cout << c << endl;
}
```

OUTPUT

```
A
65
75
B
```

5. Bảng mã ASCII và các câu lệnh kiểm tra loại ký tự

Câu lệnh	Ý nghĩa
<pre>char c; if ((c >= 'a') && (c <= 'z'))</pre>	Kiểm tra ký tự in thường
<pre>char c; if ((c >= 97) && (c <= 122))</pre>	Kiểm tra ký tự in thường
<pre>char c; if ((c >= 'A') && (c <= 'Z'))</pre>	Kiểm tra ký tự in hoa
<pre>char c; if ((c >= 65) && (c <= 90))</pre>	Kiểm tra ký tự in hoa
<pre>char c; if ((c >= '0') && (c <= '9'))</pre>	Kiểm tra ký tự là chữ số
<pre>char c; if ((c >= 48) && (c <= 57))</pre>	Kiểm tra ký tự là chữ số
<pre>char c = 'A'; c += 32;</pre>	Chuyển ký tự c thành dạng in thường tương ứng
<pre>char c = 'a'; c -= 32;</pre>	Chuyển ký tự c thành dạng in hoa tương ứng





28TECH

Become A Better Developer

You got 2 choices :

- 1. Do it now**
- 2. Regret it later**



FINISH

