Chương 4. Điều khiển CT C

1. Các yếu tố cơ bản vòng lặp đk đếm
2. Lệnh for và do ... while
3. Nhiều lựa chọn sử dụng switch
4. Sử dụng lệnh break và continue để đảo thứ tự đk
5. Sử dụng lệnh logic

1. Các yếu tố cơ bản vòng lặp đk đếm

- Vòng lặp: Nhóm lệnh thực hiện lặp lại khi điều kiện thỏa
- Vòng lặp điều khiển đếm
  - Vòng lặp có số lần lặp xác định
  - Điều khiển biến đếm số lần lặp
- Vòng lặp điều khiển bằng ký tự canh
  - Lặp không xác định
  - Được sử dụng khi số vòng lặp không biết trước
  - Giá trị canh xác định kết thúc dữ liệu

1. Các yếu tố cơ bản vòng lặp đk đếm

- Vòng lặp điều khiển đếm yêu cầu:
  - Tên của biến điều khiển (hoặc đếm vòng lặp)
  - Giá trị khởi tạo của biến điều khiển
  - Tăng hoặc giảm biến điều khiển được cập nhật mỗi lần lặp
  - Điều kiện kiểm tra giá trị cuối của biến điều khiển (có lặp tiếp hay không?)

1. Các yếu tố cơ bản vòng lặp đk đếm

- Ví dụ:

```
int counter = 1;
while ( counter <= 10 )
{
   printf( "%d\n", counter );
   ++counter;
}
```

- int counter = 1;
  - Tên biến counter, kiểu giá trị integer
  - Đặt nó vào, tạo khoảng trống trong bộ nhớ
  - Khởi giá trị ban đầu bằng 1

1. Các yếu tố cơ bản vòng lặp đk đếm

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
	unsigned int counter = 1;
	while (counter <= 10)
	{
		printf("%u\n", counter);
		++counter;
	}
	getch();
}
```

Điều chỉnh CT để hiển thị số trên 1 dòng, cách nhau 5 khoảng trống

2. Vòng lặp for

- Cấu trúc:
  - for (khởi tạo ; kiểm tra điều kiện ; tăng)
    { khối lệnh }
- VD:
  - for (int counter = 1; counter <=10; counter++)
    {
    printf (“%d\n”, counter);
    }
- Viết lại vòng lặp for thành dạng vòng lặp while

  Khởi tạo ;
  while (điều kiện tiếp tục vòng lặp)
  { lệnh;
  tăng; }

2. Vòng lặp for

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
	unsigned int counter;
	for (counter = 1; counter <= 10; ++counter)
	{
		printf("%u\n", counter);
	}
	getch();
}
```

Cách viết khác

```
Initialize counter to 0
while ( ++counter <= 10 )
    printf( “%d\n”, counter );
```

- Giải thích tại sao khởi tạo counter = 0?
- Viết CT hoàn chỉnh

Nếu có nhiều biến, có thể viết như sau:

```
for (int i = 0, j = 0;  j + i <= 10; j++, i++)
   printf( "%d\n", j + i );
```

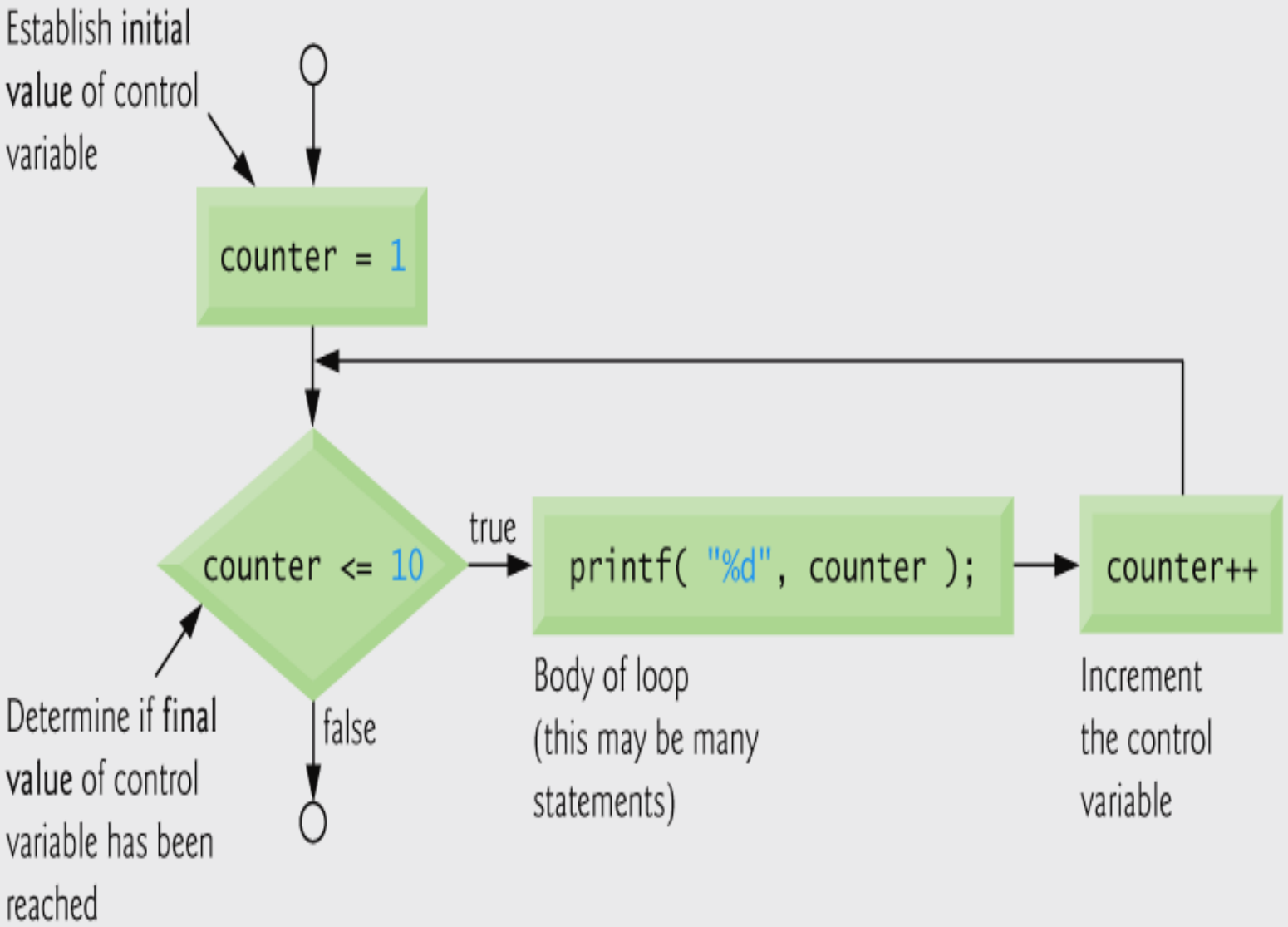
Chú ý

- Khởi tạo, điều khiển vòng lặp, tăng có thể chứa biểu thức
- VD: Nếu x = 2 và y = 10

  for (j = x; j <= 4 * x * y; j += y / x)

  Tương đương với

  for (j = 2; j <= 80; j += 5)
- Có thể tăng hoặc giảm
- Nếu điều kiện vòng lặp ban đầu là False thì thân lệnh for không thực hiện. Chương trình sẽ tiếp tục thực hiện lệnh sau for
- Biến điều khiển thường được in hoặc sử dụng trong thân nhưng không cần thiết


Establish initial value of control variable
counter = 1
counter <= 10
true
printf("%d", counter);
counter++
Body of loop (this may be many statements)
Increment the control variable
Determine if final value of control variable has been reached
false

Exercise

- 1. Viết CT tính tổng các số chẵn từ 0 đến n với n nhập từ bàn phím
- 2. Tính và in ra bảng tiền nhận được khi gởi tiết kiệm tương ứng từ 1 đến 10 năm với công thức tính như sau:

  Tiền nhận = tiền gốc*((1+ lãi mỗi năm)^ số năm)
- 3. Kết quả a, b của đoạn lệnh sau:

```
int i, a = 3;
for ( i = 0; i<5; i++)
    a ++;
```

Kết quả : a = …….. i = ……..

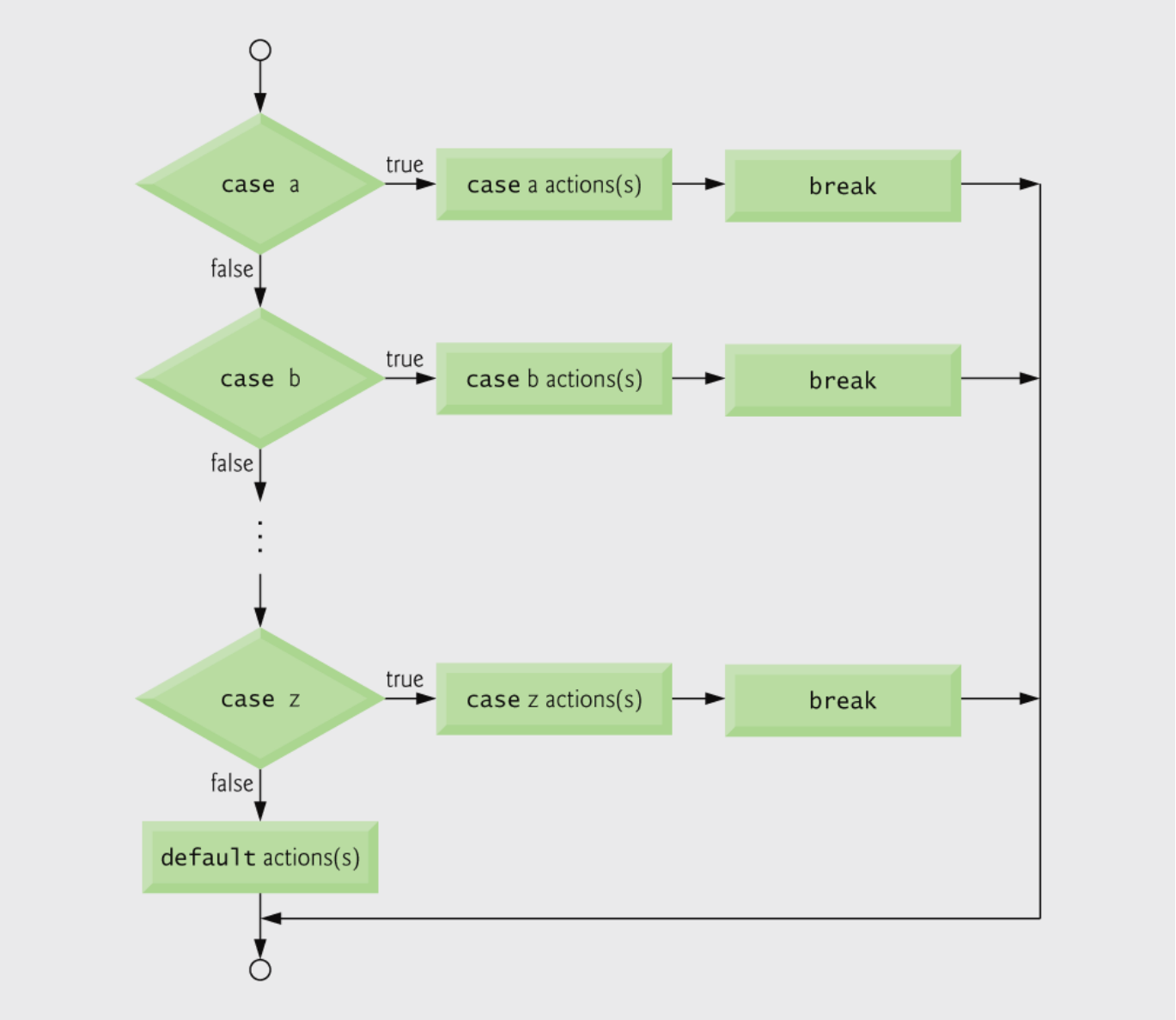
3. Lựa chọn sử dụng Switch

- Được sử dụng khi 1 biến hoặc 1 biểu thức được kiểm tra
- Cấu trúc:

```
Switch (value)
    {
        Case “1”:
            actions
        Case “2”:
            actions
        ……..
        Default:
            actions
    }
```

Dùng break; để thoát

3. Lựa chọn sử dụng Switch



VD: Đếm sl xếp loại của SV từ A → F

```
#include <stdio.h>
int main( void )
{
int grade;
unsigned int aCount = 0, bCount = 0, cCount = 0, dCount = 0, fCount = 0;
puts( "Enter the letter grades." );
puts( "Enter the EOF character to end input." );
while ((grade = getchar()) != EOF)
{
        switch (grade)
          {
              case 'A':
              case 'a':
                  ++aCount;
                  break;
              case 'B':
              case 'b':
                  ++bCount;
                  break;
```

```
case 'C':
case 'c':
        ++cCount;
        break;
case 'D':
case 'd':
        ++dCount;
        break;
case 'F':
case 'f':
        ++fCount;
        break;
case '\n':
case '\t':
case ' ':
        break;
        default:
    printf("%s", "Incorrect letter grade entered.");
puts(" Enter a new grade.");
break;
}
}
```

```
puts("\nTotals for each letter grade are:");
printf("A: %u\n", aCount);
printf("B: %u\n", bCount);
printf("C: %u\n", cCount);
printf("D: %u\n", dCount);
printf("F: %u\n", fCount);
getch();
}
```

Chú ý

- Thiếu break; gây ra lỗi logic
- EOF của các hệ thống khác nhau là khác nhau. Của Windows là CLTR+Z

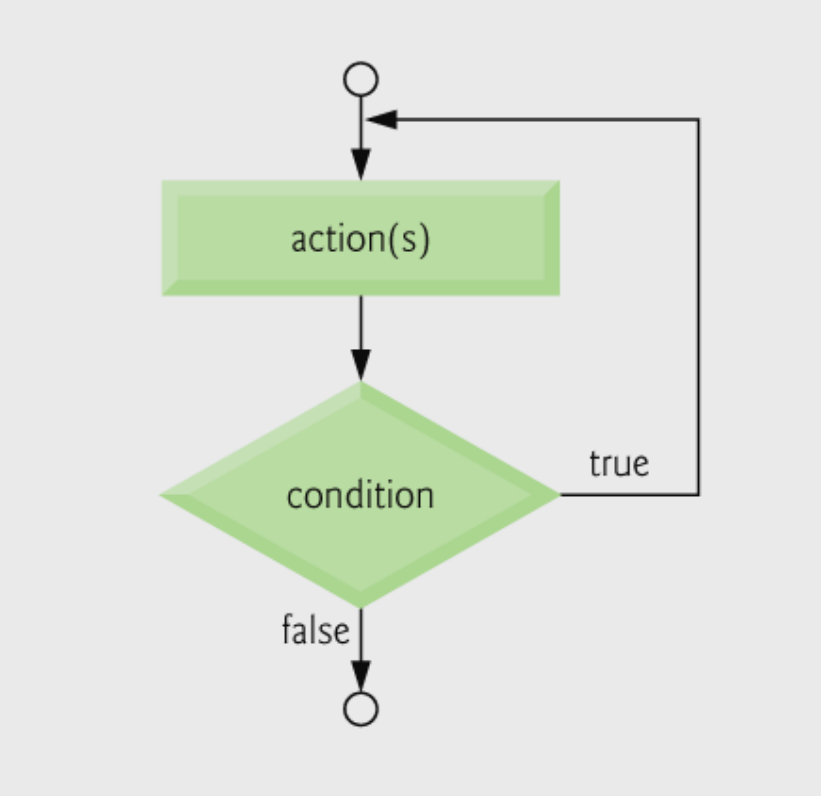
Exercise 4

```
int a=1, b=5, c=3;
switch (b-c)
{
case 0:
        a++;
        break;
case 1:
        b++;
        break;
default:
        a=++b-c;
}
```

a= …….; b= …….

4. Vòng lặp do...while

- Điều kiện vòng lặp: chỉ kiểm tra sau khi thực hiện khối lệnh
- Khối lệnh được thực hiện ít nhất 1 lần



Exercise

5. Cho biết kết quả thực hiện lệnh

```
int   i = 0;
do
{
    printf("%d",i);
}
while  ( i++ <4);
```

Kết quả : .....

Exercise

- 6. Cho biết kết quả a và i sau khi thực hiện chương trình

```
int a=5, b=2, i=0;
do
{
        a=a-b;
        i++;
}
while(i<3);
```

5. Lệnh break và continue

- Lệnh break
  - Được sử dụng để thoát vòng lặp: while, for, do ... while, switch
  - Tiếp tục thực hiện lệnh đầu tiên sau break
  - Các trường hợp sử dụng:
    - Thoát sớm khỏi vòng lặp
    - Bỏ qua phần còn lại của lệnh switch

VD

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
        int x;
        for (x = 1; x <= 10; ++x)
        {
            if (x == 5)
                { break;}
            printf("%u \n", x);
        }
        printf("\nBroke out of loop at x == %u\n", x);
        getch();
}
```

Exercise

- 7. Cho biết kết quả dòng lệnh sau:

```
int  i, a = 3;
for ( i = 0; ; i ++ )
{
    a += 2;
   if ( a >= 8 )
     break;
}
```

Kết quả : a = …….. i = …….

5. Lệnh break và continue

- Lệnh continue
  - Bỏ qua phần còn lại trong vòng lặp để tiến hành các phần tiếp theo của vòng lặp
  - VD: In các số từ 0 đến 10, bỏ qua số 5

```c
#include <stdio.h>
#include <conio.h>
void main(void)
{
   int x;
   for (x = 1; x <= 10; x++)
   {
     if (x == 5)
        {
           continue;
        }
     printf("%d ", x);
   }
   printf("\nUsed continue to skip printing the
value 5\n");
   getch();
}
```

6. Chú ý

- Toán học: + - * / %
- Biểu thức logic: && || !
- Toán tử quan hệ: == != > >= < <=
- Tăng giảm: ++i --i i++ i--
- Toán tử điều kiện:(toán hạng 1) ? (toán hạng 2) : (toán hạng 3)
- Nhầm giữa phép so sánh và phép gán:
  - Gây ra lỗi logic
  - Giá trị nonzero (#0) luôn true
    zero (=0) luôn false
  - VD: Chương trình nạp mã báo trúng thưởng hoặc không

```c
#include <stdio.h>
#include <conio.h>
int main(void)
{
    int playcode;
    printf("\n enter your code\n");
    scanf("%d", &playcode);
    if (playcode == 4)
      {
        printf("%s", "You get a bonus!");
      }
    else
      printf("%s", "goodluck next time");
    getch();
}
```

1. Cho biết kết quả khi playcode = 4 và 0
2. Playcode = 4, cho biết kết quả khi playcode =4, 2, 0
3. Playcode = 0, cho biết kết quả khi playcode = 4, 2, 0

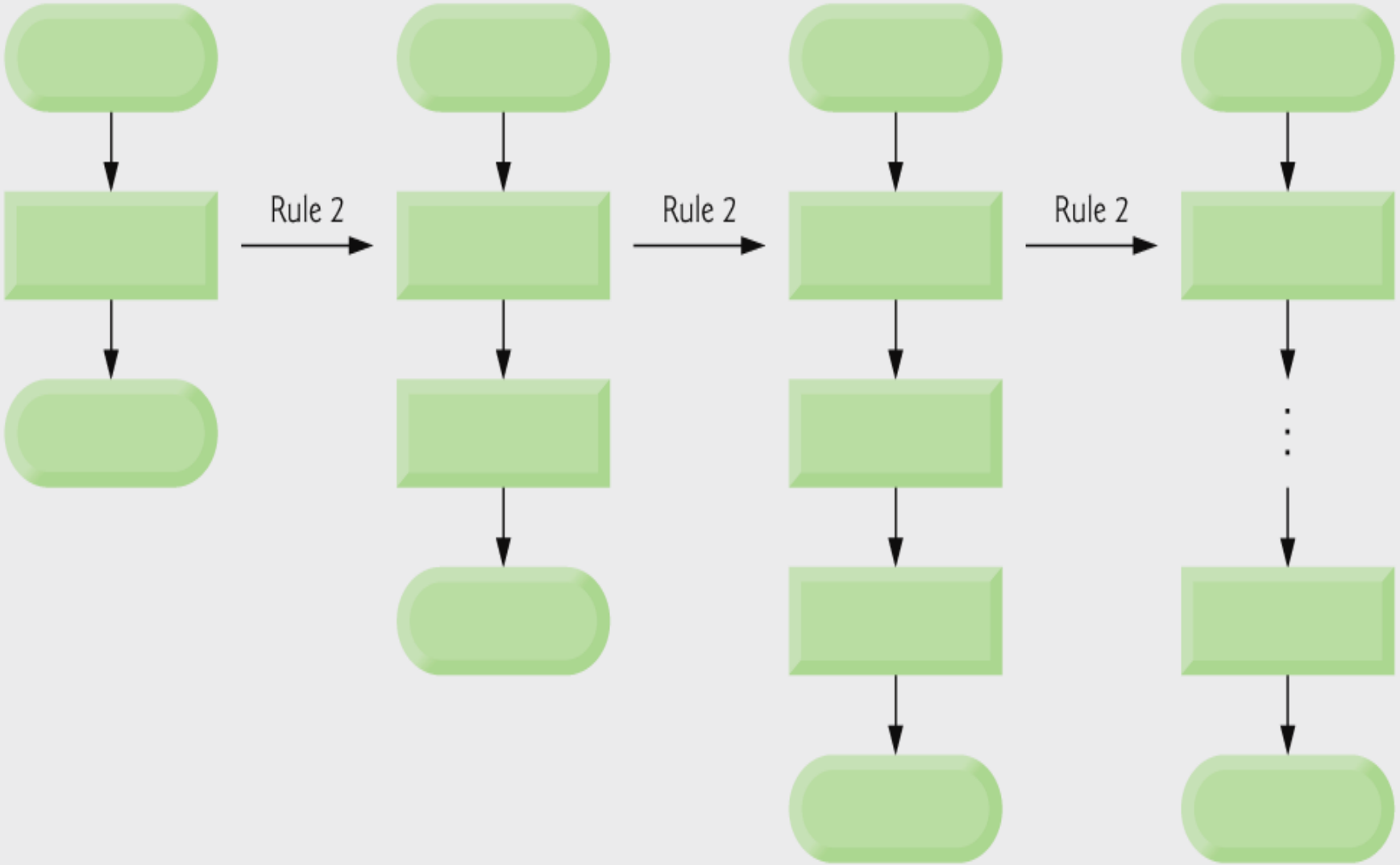
6. Chú ý

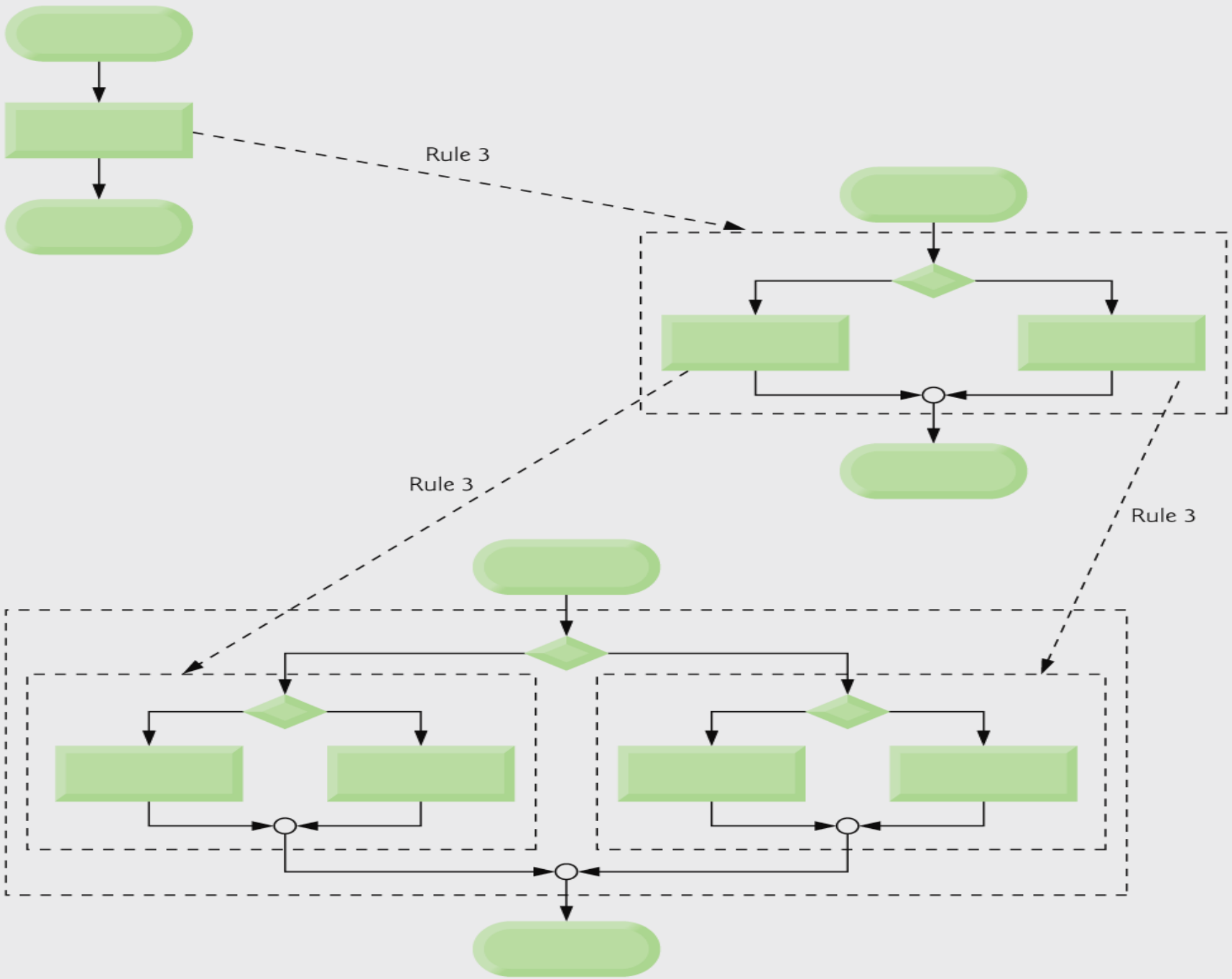
- lvalue và rvalue
  - Lvalue: là biến, được sử dụng bên trái của phép gán
  - Rvalue: hằng số, được sử dụng bên phải của phép gán
  - Lvalue được sử dụng như rvalue nhưng không ngược lại
  - Biểu thức chỉ được xuất hiện bên phải của phương trình
- VD: Chỉ có thể viết x=4 nhưng không thể viết 4 = x

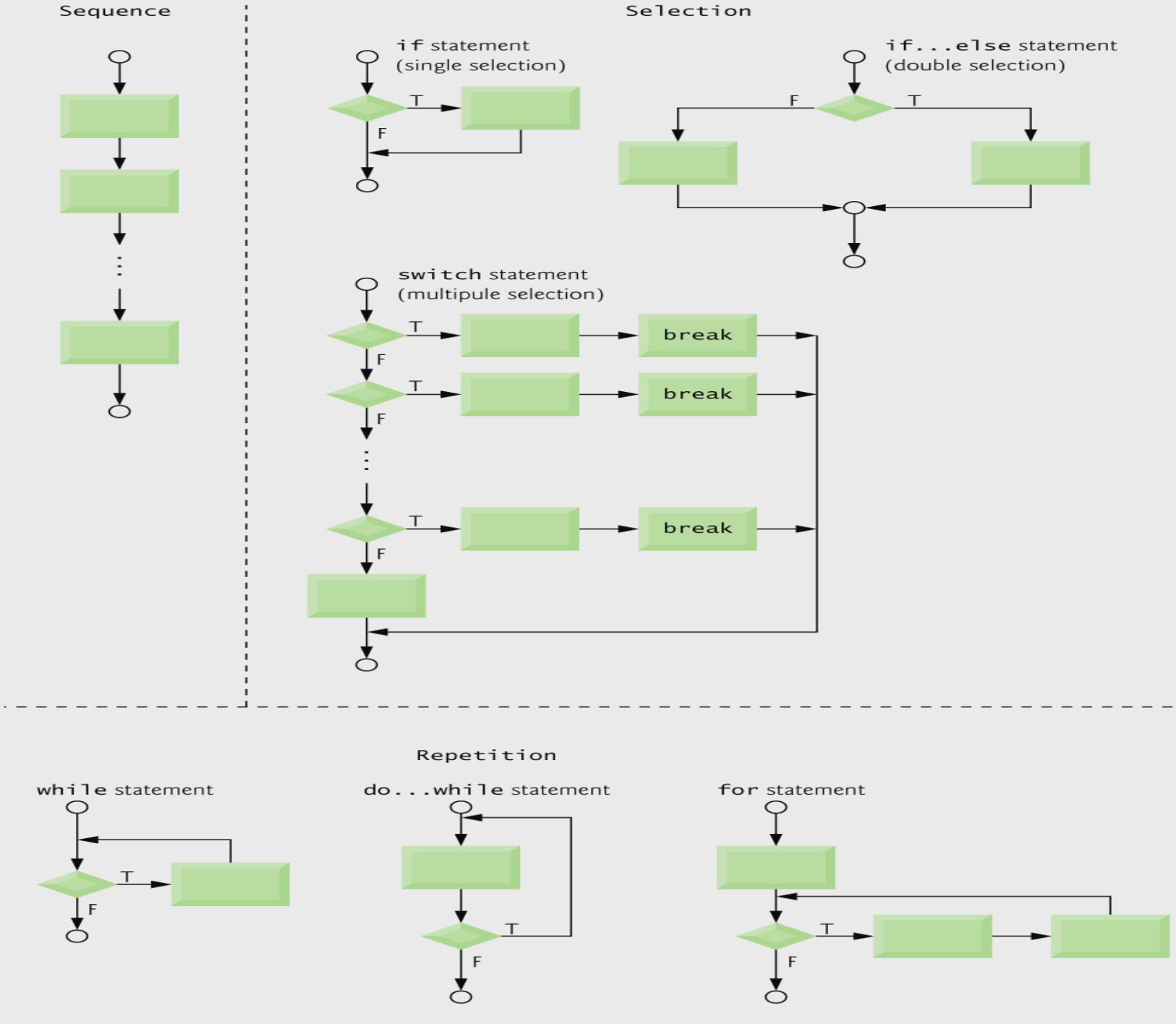
```c
#include <stdio.h>
#include <conio.h>
int main(void)
{
    int playcode;
    printf("\n enter your code\n");
    scanf("%d", &playcode);
    if (playcode == 4)
      {
        printf("%s", "You get a bonus!");
      }
    else
      printf("%s", "goodluck next time");
    getch();
}
```

Tổng kết lập trình cấu trúc

- Có cấu trúc: dễ hiểu, dễ kiểm tra, dễ sửa lỗi, dễ hiệu chỉnh
- Quy tắc hình thành CT có cấu trúc
  - Bắt đầu bằng lưu đồ đơn giản nhất
  - Đối với cấu trúc tuần tự: Cứ 1 khối được thay bằng 2,3 khối
  - Đối với cấu trúc lựa chọn hay lặp thì được thay bằng lệnh điều khiển

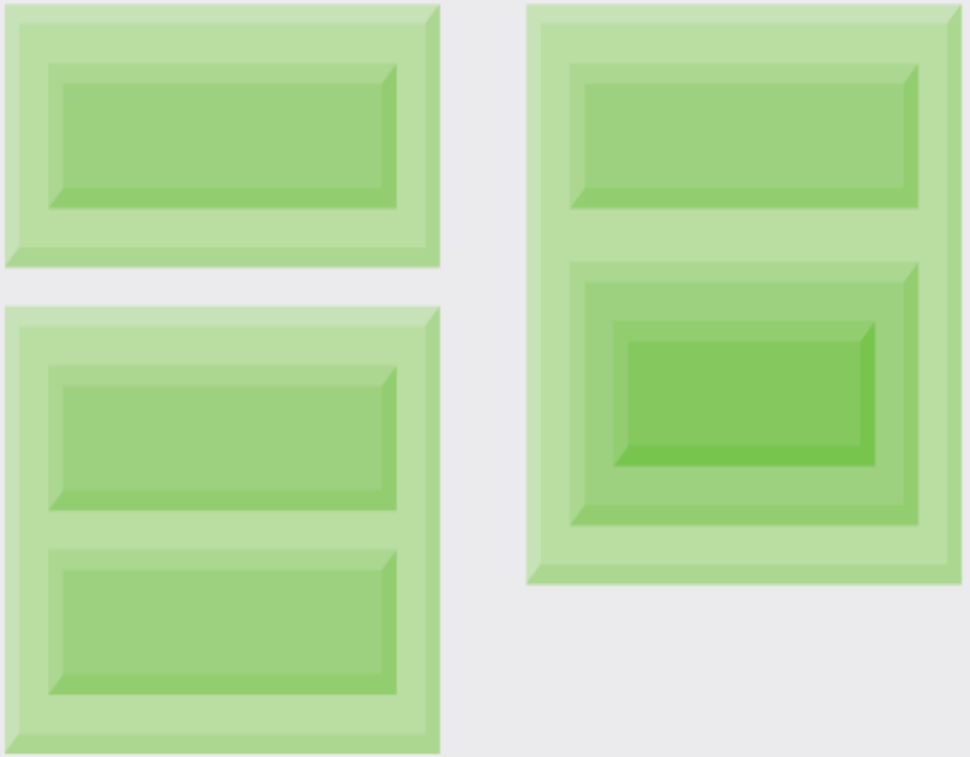

Rule 2
Rule 2
Rule 2


Rule 3
Rule 3
Rule 3

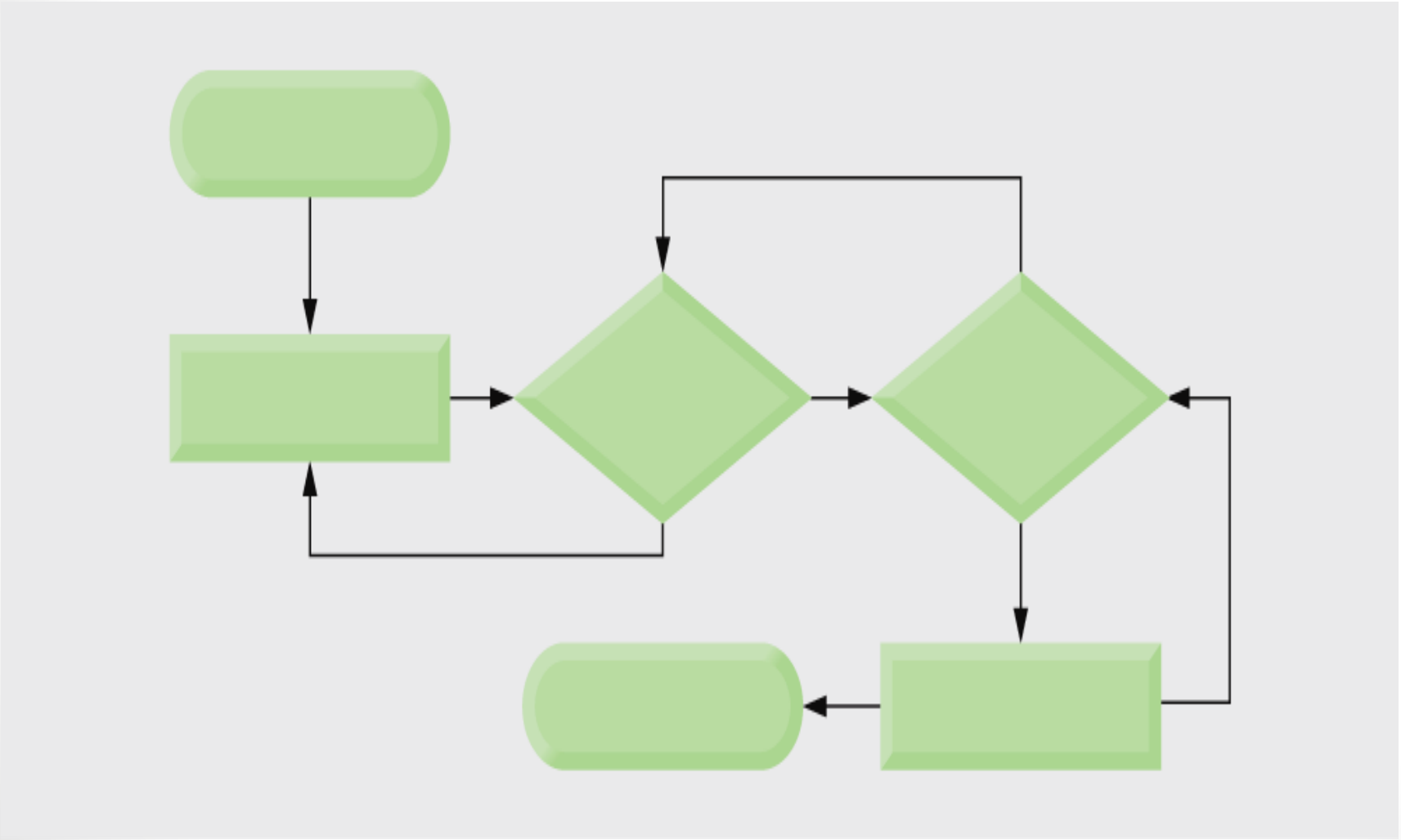

Sequence
Selection
if statement
(single selection)
T
F
if...else statement
(double selection)
F
T
switch statement
(multipule selection)
T
F
break
Repetition
while statement
do...while statement
for statement

Stacked building blocks

Nested building blocks



Overlapping building blocks
(Illegal in structured programs)



Chú ý

- Tất cả các chương trình đều có thể được viết dưới dạng cấu trúc
- Có 3 cấu trúc:
  - Tuần tự:
  - Lựa chọn: if, if...else, switch
  - Lặp: while, do...while, for
- Tất cả chương trình lựa chọn đều viết được bằng if
- Tất cả vòng lặp nào cũng được viết bằng while

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
int counter = 0;
while (counter++ <= 10)
printf("%d\n", counter);
getch();
}
```