1. In dãy số 1, 2, 3, 4, 5 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp for.
2. In dãy số 1, 3, 5, 7, 9 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp for và bằng 2 cách khác nhau.
3. In dãy số 2, 4, 6, 8, 10 … **2n** (n là số được nhập vào từ bàn phím) bằng vòng lặp for.
4. Viết chương trình tính tổng S sau đây bằng cách sử dụng vòng lặp for:

S=1 + 2 + 3 +....+ n (Số n được nhập vào từ bàn phím)

1. Viết chương trình tính tổng S sau đây bằng cách sử dụng vòng lặp for:

S=1 + 1/2 + 1/3 +....+ 1/n (Số n được nhập vào từ bàn phím)

1. In dãy số 2, -4, 6, -8, 10 … n (n là số được nhập vào từ bàn phím) bằng cách sử dụng vòng lặp for và bằng 2 cách khác nhau.

**Gợi ý:** cách 1 dùng toán tử %, cách 2 dùng 1 biến để xác định dấu âm và dấu dương. (\*)

1. Tính giai thừa của số n (n là số được nhập vào từ bàn phím) bằng cách sử dụng vòng lặp for.
2. In dãy số 1, 2, 3, 4, 5 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp **while**.
3. In dãy số 1, 3, 5, 7, 9 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp **while** và bằng 2 cách khác nhau.
4. In dãy số 1, 2, 3, 5, 8, 13, …n (n là số được nhập vào từ bàn phím) bằng vòng lặp **while**.
5. In dãy số 1, 3, 7, 15, 31, …n (n là số được nhập vào từ bàn phím) bằng vòng lặp **while**.
6. In dãy số 1, -3, 7, -15, 31, … (n là số được nhập vào từ bàn phím) bằng vòng lặp **while**.
7. Tính tổng và tích các chữ số của một số m, m được nhập từ bàn phím:

(Ví dụ : m = 234 => S = 2 + 3 + 4 = 9, P = 2 \* 3 \* 4 = 24)

**Gợi ý:** học viên tự suy nghĩ nên sử dụng vòng lặp for hay while để làm bài này.

1. Nhập 2 số nguyên dương a và b. Sau đó in ra ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của 2 số nguyên dương a và b đó.

**Gợi ý:** học viên tự suy nghĩ nên sử dụng vòng lặp for hay while để làm bài này.

1. Cho đoạn code sau. Hãy viết kết quả ra giấy (không được dùng trình biên tập, sau khi viết kết quả ra giấy xong thì dùng trình biên tập để tự kiểm tra lại kết quả):

for (int i = 1; i <= 10; i++)

for (int j = 2; j <= 20; j++)

System.out.println("i=" + i + " j=" + j);

1. Cho những đoạn code sau. Hãy viết kết quả của những đoạn code này ra giấy (không được dùng trình biên tập, sau khi viết kết quả ra giấy xong thì dùng trình biên tập để tự kiểm tra lại kết quả):

**Đoạn 1:**

for (int i = 1; i <= 10; i++)

for (int j = 2; j <= 20; j++)

System.out.println("i=" + i + " j=" + j);

**Đoạn 2:**

for (int i = 1; i <= 10; i++) {

for (int j = 20; j >= 1; j--)

System.out.println("i=" + i + " j=" + j);

}

**Đoạn 3:**

for (int i = 1; i <= 10; i++) {

for (int j = 2; j <= 20; j=j+2) {

System.out.println("i=" + i + " j=" + j);

}

}

**Đoạn 4:**

for (int i = 1; i <= 10; i++)

for (int j = 2; j <= 20; j=j+2)

for (int k = 3; k <= 30; k=k+3)

System.out.printnl("i=" + i + " j=" + j + " k=" + k);

Sau đó, hãy rút ra 2 kết luận:

* Thứ nhất: là về cách chạy của các vòng lặp for lồng vào nhau.
* Thứ hai: là về việc sử dụng dấu { sau mỗi câu lệnh for

1. Viết chương trình tính:

S = 1+1/2!+1/3!+…..+1/n! (Số n được nhập từ bàn phím)

**Gợi ý:** sử dụng 2 vòng lặp for lồng nhau.

1. Viết chương trình tính:

S = 1+1/3!+1/5!+…..+1/(2n-1)! (Số n được nhập từ bàn phím)

**Gợi ý:** sử dụng 2 vòng lặp for lồng nhau.

1. Viết chương trình in ra hình vuông với những chữ A (hình vuông này có kích thước không cố định, n là số ký tự của mỗi cạnh hình vuông, n được nhập vào từ bàn phím):

Ví dụ nếu người dùng nhập n = 5, thì chương trình sẽ hiển thị hình vuông là:

AAAAA

AAAAA

AAAAA

AAAAA

AAAAA

**Gợi ý:** sử dụng 2 vòng lặp for lồng nhau.

1. Viết chương trình in ra hình vuông với những chữ A và những chữ B (hình vuông này có kích thước không cố định, n là số ký tự của mỗi cạnh hình vuông, n được nhập vào từ bàn phím):

Ví dụ nếu người dùng nhập n = 5, thì chương trình sẽ hiển thị hình vuông là:

AAAAA

BBBBB

AAAAA

BBBBB

AAAAA

Ví dụ nếu người dùng nhập n = 6, thì chương trình sẽ hiển thị hình vuông là:

AAAAA

BBBBB

AAAAA

BBBBB

AAAAA

BBBBB

**Gợi ý:** sử dụng 2 vòng lặp for lồng nhau.

1. Viết chương trình in ra hình vuông với những chữ A và những chữ B (hình vuông này có kích thước không cố định, n là số ký tự của mỗi cạnh hình vuông, n được nhập vào từ bàn phím):

Ví dụ nếu người dùng nhập n = 5, thì chương trình sẽ hiển thị hình vuông là:

ABABA

ABABA

ABABA

ABABA

ABABA

Ví dụ nếu người dùng nhập n = 6, thì chương trình sẽ hiển thị hình vuông là:

ABABAB

ABABAB

ABABAB

ABABAB

ABABAB

ABABAB

**Gợi ý:** sử dụng 2 vòng lặp for lồng nhau.

1. In các hình \* sau đây: (chiều dài các cạnh là k hông cố định)

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*

\*

\*

\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*

\*

\*

\*

\*\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*\*\*\*\*\*

1. In các hình \* sau đây: (chiều dài các cạnh là không cố định)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

\*\*\*\*\*

\* \*

\* \*

\*\*

\*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

1. In các hình \* sau đây: (chiều dài các cạnh là không cố định)

\*\*\*\*\*\*

\*

\*

\*

\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*

\*

\*

\*

\*\*\*\*\*\*

\*\*\*\*\*\*

\* \* \*

\* \* \*

\* \* \*

\*\*\*\*\*\*

\*\*\*\*\*\*

\*\* \*

\* \* \*

\* \* \*

\* \*\*

\*\*\*\*\*\*

1. Hãy chuyển đoạn code sử dụng 2 vòng lặp for sau đây thành đoạn code sử dụng 2 vòng lặp while:

**for (int i = 1; i <= n; i++) {**

**for (int j = 1; j <= n; j++) {**

**System.out.println(i + " - " + j);**

**}**

**}**

1. In hình \* sau đây: (chiều dài các cạnh là không cố định)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Yêu cầu:** Sử dụng các vòng lặp **while** lồng nhau.

1. In hình \* sau đây: (chiều dài các cạnh là không cố định)

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

**Yêu cầu:** Sử dụng các vòng lặp **while** lồng nhau.

1. In dãy số 1, 2, 3, 4, 5 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp **do-while**.
2. In dãy số 1, 3, 5, 7, 9 … n (n là số được nhập vào từ bàn phím) bằng vòng lặp **do-while**.
3. In dãy số 1, 2, 3, 5, 8, 13, …n (n là số được nhập vào từ bàn phím) bằng vòng lặp **do-while**.
4. Hãy chuyển đoạn code sử dụng 2 vòng lặp for sau đây thành đoạn code sử dụng 2 vòng lặp **do-while**:

**for (int i = 1; i <= n; i++) {**

**for (int j = 1; j <= n; j++) {**

**System.out.println(i + " - " + j);**

**}**

**}**

1. In hình \* sau đây: (chiều dài các cạnh là không cố định)

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Yêu cầu:** Sử dụng các vòng lặp **do-while** lồng nhau.

1. In hình \* sau đây: (chiều dài các cạnh là không cố định)

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

**Yêu cầu:** Sử dụng các vòng lặp **do-while** lồng nhau.

1. Hãy phân tích sự khác nhau giữa **break** và **continue** ở 2 đoạn code sau đây:

**Đoạn code 1:**

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i > 6) {

break;

}

System.out.print(i);

}

}

**Đoạn code 2:**

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i > 6) {

continue;

}

System.out.print(i);

}

}

1. Viết chương nhập vào một số nguyên dương n. Sau đó quy đổi n ra hệ nhị phân. Yêu cầu: Không được sử dụng hàm có sẵn của Java, hãy tự thiết kế thuật toán. (\*)

**Gợi ý:** học viên tự suy luận loại vòng lặp cần sử dụng để làm bài tập này.

1. Viết chương nhập vào một số nguyên a. Sau đó kiểm tra số nguyên a này có phải là số nguyên dương hay không (tức là kiểm tra xem a > 0 hay không). Nếu a không phải là số nguyên dương thì chương trình **hãy yêu cầu người dùng nhập lại a cho đến khi nào a là số nguyên dương thì thôi**. Sau khi nhập được a > 0 rồi thì thông báo “Đã nhập thành công”. (\*)

**Gợi ý:** học viên làm bằng 2 cách. Cách 1 dùng vòng lặp while, cách 2 dùng vòng lặp do-while. Sau đó tự suy luận xem: với những bài **như thế nào** thì ta nên dùng vòng lặp **while?** Những bài **như thế nào** thì ta nên vòng lặp **do-while**? Khi nào thì ta nên dùng **break** trong vòng lặp?

1. In dãy *số 2, 4, -6, 8,* 10, -12 … n (bằng 2 cách khác nhau) (n là số được nhập vào từ bàn phím) (\*)

**Gợi ý:** học viên tự suy luận loại vòng lặp và số lượng vòng lặp cần sử dụng để làm bài tập này.

1. In dãy số 2, -4, 6, 8, -10, 12, 14, 16, -18, ….., n (\*\*)

**Gợi ý:** học viên tự suy luận loại vòng lặp và số lượng vòng lặp cần sử dụng để làm bài tập này.

**(\*)(\*\*): Bài khó**

1. Vòng for sau đây chứa **bao nhiêu câu lệnh** bên trong?

for (int j = 1; j <= n; j++) {

if (j % 2 != 0) {

System.out.print("A");

} else {

System.out.print("B");

}

}