
LẬP TRÌNH CƠ BẢN VỚI JAVA

Câu hỏi và bài tập tuần 1

- Điều gì xảy ra nếu trong tệp [Hello.java](#) bạn bỏ qua:
 - `main`
 - `String`
 - `Hello`
 - `System.out`
 - `println`
- Mô tả điều gì xảy ra nếu trong tệp [Hello.java](#), bạn bỏ qua:
 - dấu `;`
 - dấu nháy kép đầu tiên `"`
 - dấu nháy kép thứ hai `"`
 - dấu mở ngoặc đầu tiên `{`
 - dấu mở ngoặc thứ hai `{`
 - dấu đóng ngoặc đầu tiên `}`
 - dấu đóng ngoặc thứ hai `}`
- Mô tả điều gì xảy ra nếu trong tệp [Hello.java](#), bạn gõ nhầm (ví dụ, quên mất kí tự thứ hai) các từ

-
- a. main
 - b. String
 - c. Hello
 - d. System.out
 - e. println

4. Gõ tệp nguồn sau, khi biên dịch không báo lỗi, nhưng khi chạy chương trình thì thấy báo lỗi `java.lang.NoSuchMethodError`. Bạn hãy thử giải thích vì sao

```
1 public class Hello {  
2     public static void main() {  
3         System.out.printf("Doesn't execute.");  
4     }  
5 }
```

Hello.java

5. Viết chương trình [TenHelloWorlds.java](#) để in ra màn hình 10 lần dòng chữ "Hello, World"
6. Sửa chương trình [UseArgument.java](#) thành [UseThree.java](#) cho phép lấy ba xâu tên trên đối dòng lệnh và in ra màn hình một câu có chứa ba tên này trong đó thứ tự các tên được in theo chiều ngược lại. Ví dụ chạy "java UseThree An Bình Chi" sẽ in ra màn hình là câu "Xin chào Chi, Bình và An".

```
1 /* Use command-line arguments for input strings */  
2 public class UseArgument {  
3     public static void main(String[] args) {  
4         System.out.print("Xin chào ");  
5         System.out.print(args[0]);  
6         System.out.println(". Bạn khỏe không?");  
7     }  
8 }
```

UseArgument.java

7. Điều gì xảy ra nếu bạn biên dịch chương trình `LeapYear.java` sau và chạy chương trình với các câu lệnh sau:

- `java LeapYear`
- `java LeapYear 1975.5`
- `java LeapYear -1975`
- `java LeapYear 1975 1976 1977`

```
1 public class LeapYear {
2     public static void main(String[] args) {
3         int year = Integer.parseInt(args[0]);
4         boolean isLeapYear;
5
6         // divisible by 4
7         isLeapYear = (year % 4 == 0);
8
9         // divisible by 4 and not 100
10        isLeapYear = isLeapYear && (year % 100 != 0);
11
12        // divisible by 4 and not 100 unless divisible
13        by 400
14        isLeapYear = isLeapYear || (year % 400 == 0);
15
16        System.out.println(isLeapYear);
17    }
18 }
```

LeapYear.java

8. Viết chương trình `DivideByZero.java` để kiểm tra xem điều gì xảy ra nếu chia một số `int` hoặc số `double` cho 0.
9. Viết chương trình cho phép đọc 3 tham số từ dòng lệnh và trả về kết quả là `true` nếu ba giá trị bằng nhau và là `false` nếu ngược lại.
10. Viết chương trình `Ordered.java` cho phép nhập 3 số nguyên `x`, `y`, `z` là tham số dòng lệnh. Tạo biến kiểu `boolean` nhận giá trị là

`true` nếu 3 số nguyên được sắp theo thứ tự tăng dần hoặc giảm dần, và bằng `false` nếu ngược lại. In ra màn hình biến `b` .

11. Viết chương trình cho phép nhập 4 số thực `x1`, `x2`, `y1`, `y2` từ dòng lệnh và in ra khoảng cách Euclide giữa điểm `(x1, y1)` và điểm `(x2, y2)`. Sử dụng hàm `Math.sqrt()`.
12. Viết chương trình [Divisibility.java](#) cho phép đọc 2 biến là tham số dòng lệnh và in ra giá trị `true` nếu cả hai đều chia hết cho 7 và `false` nếu ngược lại.
13. **Sắp xếp.** Viết chương trình [ThreeSort.java](#) nhằm lấy 3 giá trị nguyên từ dòng lệnh và in 3 giá trị này ra màn hình theo thứ tự tăng dần. Sử dụng hàm `Math.min(int a, int b)` và `Math.max(int a, int b)`.

Câu hỏi và bài tập tuần 2

1. Giả sử a , b là hai biến kiểu `int`. Khối lệnh sau thực hiện công việc gì ?

```
int t = a;  
b = t;  
a = b;
```

2. Giả sử a , b là các biến kiểu `int`. Biểu thức sau có ý nghĩa gì?
(!($a < b$) && !($a > b$))

3. Tại sao kết quả của phép toán $10/3$ lại là 3 mà không phải 3.3333333

4. Một sinh viên vật lý thu được kết quả không đúng khi sử dụng đoạn chương trình sau để tính giá trị F theo công thức $F = Gm_1m_2/r^2$

```
double force = G * mass1 * mass2 / r*r
```

Hãy sửa lại đoạn chương trình để thu được kết quả đúng.

5. Vì sao câu lệnh sau mắc lỗi cú pháp nếu `grade` là biến có kiểu `int`. Hãy sửa lại câu lệnh cho đúng?

```
boolean isA = (90 <= grade <= 100);
```

6. Giá trị trả về của biểu thức sau là gì?

```
( Math.sqrt(2) * Math.sqrt(2) ==2)
```

7. Chương trình dịch sẽ làm gì nếu bạn viết câu lệnh sau:

```
int a = 27 * "three";
```

8. Chương trình dịch sẽ làm gì nếu bạn viết câu lệnh sau:

```
double x;  
System.out.println(x);
```

Lưu ý về phần bài tập lập trình

- Tất cả các dữ liệu đầu vào đều được nhập qua đối dòng lệnh
- Tất cả các chức năng được yêu cầu trong mỗi chương trình phải được viết thành chương trình con riêng (là hàm), hàm `main` chỉ làm việc nhận giá trị từ đối dòng lệnh và gọi tới các chương trình con.

1. Viết chương trình `Distance.java` cho phép nhập hai số nguyên x , y và in ra màn hình khoảng cách Euclid từ điểm (x, y) đến điểm gốc toạ độ $(0, 0)$.
2. Viết lại chương trình `TenHelloWorlds.java` thành `HelloWorld.java` cho phép in ra màn hình n lần dòng chữ "Hello, World".
3. **Hàm mũ** Giả sử x là biến dương kiểu `double`. Viết chương trình `Exp.java` để tính giá trị của e^x sử dụng khai triển Taylor sau, với n cho trước:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!}$$

4. **Các hàm lượng giác.** Viết hai chương trình `Sin.java` và `Cos.java` để tính $\sin x$ và $\cos x$ sử dụng khai triển Taylor sau, với sai số ϵ cho trước (theo nghĩa trị tuyệt đối của toán hạng tiếp theo trong chuỗi nhỏ hơn ϵ):

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

5. **Sức gió.** Cho nhiệt độ t (độ F) và tốc độ gió v (đơn vị dặm/giờ), đơn vị dự báo thời tiết định nghĩa ảnh hưởng của nhiệt độ (sức gió) như sau:

$$w = 35.74 + 0.6215t + (0.4275t - 35.75)v^{0.16}$$

Viết chương trình [WindChill.java](#) để in ra giá trị sức gió. Sử dụng `Math.pow(a, b)` để tính a^b .

Chú ý: Công thức sẽ không chính xác trong trường hợp giá trị tuyệt đối của t lớn hơn 50 hoặc v lớn hơn 130 hoặc v nhỏ hơn 3

6. **Ngày trong tuần.** Viết chương trình [DayOfWeek.java](#) lấy ngày, tháng, năm là dữ liệu đầu vào và in ra thứ tương ứng của ngày đó là thứ mấy trong tuần. Chương trình lấy ba tham số: m (tháng), d (ngày) và y (năm), trong đó m bằng 1 tương ứng Tháng Một, 2 tương ứng Tháng 2 v.v. Dữ liệu đầu ra in ra giá trị bằng 0, 1, 2, 3, 4, 5, 6 tương ứng Thứ Hai, Thứ Ba, Thứ Tư, Thứ Năm, Thứ Sáu, Thứ Bảy, Chủ Nhật. Sử dụng công thức sau trong đó kí hiệu $/$ là kí hiệu của phép chia nguyên.

$$y_0 = y - (14 - m)/12$$

$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 \times ((14 - m)/12) - 2$$

$$d_0 = (d + x + 31m_0/12) \bmod 7$$

Ví dụ: Ngày mùng 2 tháng 8 năm 1953 là thứ mấy?

$$y_0 = 1953 - 0 = 1953$$

$$x = 1953 + 1953/4 - 1953/100 + 1953/400 = 2426$$

$$m_0 = 8 + 12 * 0 - 2 = 6$$

$$d_0 = (2 + 2426 + (31 * 6)/12) \bmod 7 = 2443 \bmod 7 = 0$$

$d_0 = 0$ tức là ngày 2 tháng 8 năm 1953 là ngày Thứ Hai

Bài tập tuần 3

Viết lớp (chương trình) có tên `NumberProperty`, trong lớp này có các hàm sau:

1. hàm `prime` để kiểm tra một số cho trước có phải là số nguyên tố không?
2. (*) hàm `palindrome` để kiểm tra một số cho trước có phải là số palindrome (đối xứng) hay không?
3. (*) hàm `armstrong` để kiểm tra một số có phải là số Armstrong hay không?
4. hàm `main`

Yêu cầu: In ra màn hình

1. n số nguyên tố đầu tiên
2. (*) n số palindrome đầu tiên
3. (*) n số palindrome đầu tiên là số nguyên tố
4. (*) n số Armstrong đầu tiên

Trong đó n là đối dòng lệnh

Chú ý

1. Một số/xâu được gọi là *palindrome* nếu ta viết số hay xâu đó theo thứ tự ngược lại ta thu được đúng số/xâu đó.

- 15 số *palindrome* đầu tiên: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55

- 15 số *palindrome* đầu tiên là số nguyên tố là:

2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 373, 383, 727

2. Một số *Armstrong* có n chữ số là số có giá trị đúng bằng tổng lũy thừa n của mỗi chữ số của nó.

- 371 là một số *Armstrong* có 3 chữ số vì

$$3^3 + 7^3 + 1^3 = 371$$

- 1634 là một số *Armstrong* có 4 chữ số vì

$$1^4 + 6^4 + 3^4 + 4^4 = 1634$$

- Ví dụ một số *Armstrong* là:

1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727

Câu hỏi và bài tập tuần 4

1. Mô tả và giải thích điều gì xảy ra khi bạn biên dịch chương trình có câu lệnh sau:

```
int[] a = new int[-17];
```

2. Mô tả và giải thích điều gì xảy ra khi bạn biên dịch chương trình HugeArray.java trong đó có các câu lệnh sau:

```
int n = 1000;  
int[] a = new int[n*n*n*n];
```

3. Khối lệnh sau đúng hay sai? Vì sao?

```
int[] a;  
for (int i = 0; i < 10; i++)  
    a[i] = i * i;
```

4. Khối lệnh sau thực hiện in ra màn hình cái gì? Vì sao?

```
int[] a = {1, 2, 3};  
int[] b = {1, 2, 3};  
System.out.println(a==b);
```

5. Giả sử `b[]` là mảng gồm 100 phần tử, với các giá trị ban đầu bằng 0 và `a[]` là mảng gồm N phần tử, giá trị các phần tử trong mảng là các số nguyên nằm trong khoảng từ 0 đến 99. Vòng lặp sau thực hiện việc gì?

```
for (int j = 0; j < N; j++)  
    b[a[j]] ++;
```

Lưu ý về phần bài tập lập trình

- Tất cả các dữ liệu đầu vào đều được nhập theo cách lấy từ tệp văn bản

-
- Tất cả các chức năng được yêu cầu trong mỗi chương trình phải được viết thành chương trình con riêng (là hàm), hàm main chỉ làm việc nhận giá trị từ đối dòng lệnh và gọi tới các chương trình con.

1. Viết chương trình nhập vào 2 mảng số nguyên a, b gồm n phần tử. Tính và in ra màn hình mảng c là tổng của a và b : $c[i] = a[i] + b[i]$ với mọi $i = 1, \dots, n$.
2. Viết chương trình nhập vào mảng số nguyên a gồm n phần tử, đếm và in ra màn hình số các số nguyên tố trong mảng a .
3. (*) Cho ma trận nguyên $a[][]$ kích cỡ $m \times n$, kiểm tra nếu phần tử $a[i][j]$ có giá trị bằng 0, thì chuyển các giá trị tại hàng i và cột j của ma trận a thành 0.

Chú ý: Không được sử dụng thêm bất kì mảng ngoài nào khác.

4. (*) Sinh ngẫu nhiên các giá trị của mảng nguyên 2 chiều có kích thước $m \times n$. Viết chương trình `Spiral.java` để in ra màn hình các giá trị của mảng theo thứ tự hình xoắn ốc.

Ví dụ với mảng hai chiều sau:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Kết quả sau khi chạy chương trình thu được dãy:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

5. (*) Nhập các phần tử của một mảng nguyên 2 chiều có kích thước 9×9 , giá trị các phần tử của mảng nằm trong đoạn từ 1 đến 9. Kiểm tra xem mảng này có phải là nghiệm đúng của trò chơi Sudoku hay không? Nghiệm đúng của trò chơi Sudoku là nghiệm sao cho mỗi

cột, mỗi hàng, và mỗi phần trong số chín khối con kích thước 3×3 cấu tạo nên khối chính kích thước 9×9 đều chứa tất cả các chữ số từ 1 tới 9.

5	3	4		6	7	8		9	1	2
6	7	2		1	9	5		3	4	8
1	9	8		3	4	2		5	6	7
-----+-----+-----										
8	5	9		7	6	1		4	2	3
4	2	6		8	5	3		7	9	1
7	1	3		9	2	4		8	5	6
-----+-----+-----										
9	6	1		5	3	7		2	8	4
2	8	7		4	1	9		6	3	5
3	4	5		2	8	6		1	7	9

Bài tập tuần 5

Lưu ý về phần bài tập lập trình

- Trừ khi có yêu cầu sinh ngẫu nhiên dữ liệu, tất cả các dữ liệu đầu vào đều được nhập theo cách điều hướng đầu vào chuẩn từ tệp văn bản.
- Tất cả các chức năng được yêu cầu trong mỗi chương trình phải được viết thành chương trình con riêng (là hàm), hàm `main` chỉ làm việc nhận giá trị từ đối dòng lệnh và gọi tới các chương trình con.
- Các bài tập có đánh dấu (*) ở trước là các bài tập nâng cao.

1. Viết chương trình nhập vào một mảng số nguyên a gồm n phần tử. Tính và in ra các giá trị sau: giá trị nhỏ nhất, giá trị lớn nhất, giá trị trung bình (in 2 chữ số phần thập phân) của các phần tử của mảng. Yêu cầu viết hàm (phương thức) riêng cho việc tính mỗi giá trị trên, cụ thể là:

- Hàm tính giá trị nhỏ nhất có dạng sau:

```
public static int min(int[] arr) {  
    // Statements  
}
```

- Hàm tính giá trị lớn nhất có dạng sau:

```
public static int max(int[] arr) {  
    // Statements  
}
```

- Hàm tính giá trị trung bình có dạng sau:

```
public static double average(int[] arr) {  
    // Statements  
}
```

Ví dụ dữ liệu đầu vào:

5 5 5 9 8 10

Kết quả đầu ra:

min = 5

max = 10

average = 7.40

2. Viết chương trình nhập vào dãy số nguyên a gồm n phần tử. Viết các hàm (phương thức) cho phép kiểm tra các tính chất sau của dãy:

- Dãy tăng ($a_i < a_{i+1} \forall i$) (*increasing sequence*)
- (*)Dãy đơn điệu (dãy tăng hoặc giảm) (*monotonic sequence*)
- (*)Dãy là cấp số cộng (hiệu của một số với số đứng ngay sau nó trong dãy là một hằng số) (*arithmetic sequence*). Trong trường hợp đó, in ra công thức tính tổng dãy và kết quả tính tổng dãy.
- (*)Dãy là cấp số nhân (thương của một số với số đứng ngay sau nó trong dãy là một hằng số khác không) (*geometric sequence*). Trong trường hợp đó, in ra công thức tính tổng dãy và kết quả tính tổng dãy.

Ví dụ dữ liệu đầu vào:

5 2 4 8 16 32

Kết quả đầu ra:

Dãy tăng

(*)Dãy đơn điệu

(*)Dãy cấp số nhân

(*)Tổng dãy = $2 * (\text{pow}(2, 5) - 1) / (2 - 1) = 62$

-
3. Viết chương trình nhập vào dãy số nguyên a gồm n phần tử. Viết hàm (phương thức) xoá đi phần tử ở vị trí chính giữa dãy nếu số phần tử trong dãy là lẻ. In dãy kết quả thu được.

Ví dụ dữ liệu đầu vào:

5 2 4 8 16 32

Kết quả đầu ra:

2 4 16 32

4. Viết chương trình nhập vào dãy số nguyên a gồm n phần tử.
- Viết hàm (phương thức) sắp xếp các phần tử của dãy theo thứ tự tăng dần. In ra kết quả sắp xếp.
 - (*)Tìm và in ra giá trị nhỏ nhất (min), lớn nhất (max), trung bình (average), trung vị (median) và phương sai (variance) của dãy. (Nếu có các khái niệm chưa học thì sinh viên tự tìm hiểu các khái niệm này).

Mã giả thuật toán sắp xếp nổi bọt:

Thuật toán 0.1 Sắp xếp nổi bọt BubbleSort

```
Require: array  $a$ , size  $N$ 
for all  $i$  from 0 to  $N - 1$  do
  for all  $j$  from 0 to  $N - i - 2$  do
    if  $a[j] > a[j+1]$  then
      swap(  $a[j]$ ,  $a[j + 1]$  )
    end if
  end for
end for
```

Sinh viên tự tìm đọc tài liệu về các thuật toán sắp xếp khác từ đơn giản như sắp xếp chèn (*insertion sort*), sắp xếp chọn (*selection sort*), đến phức tạp hơn một chút như sắp xếp trộn (*merge sort*), sắp xếp nhanh (*quick sort*), sắp xếp vun đống (*heap sort*).

-
5. Viết chương trình nhập vào 2 mảng số thực tương ứng với 2 véc tơ n chiều. Tính và in ra tích vô hướng, tích có hướng trong trường hợp $n = 3$ của 2 véc tơ trên.
6. Nhập kích thước m, n rồi sinh ngẫu nhiên các giá trị của hai ma trận nguyên $a[][]$ kích thước $m \times n$ và $b[][]$ kích thước $n \times m$. Tính và in ra tích 2 ma trận.
7. (*) Nhập kích thước n rồi sinh ngẫu nhiên các giá trị của một ma trận vuông $a[][]$ kích thước $n \times n$. Viết phương thức cho phép chuyển vị ma trận a (không tạo thêm ma trận mới).
8. (*) Viết chương trình chia bộ bài tự động cho 4 người, mỗi người 8 quân theo cách sau:
- Tạo mảng lưu 52 quân bài $\{2-10, J, Q, K, A\} \times \{Cơ, Rô, Bích, Tép\}$;
 - Tráo bài (bằng cách duyệt từng phần tử mảng và đổi chỗ với một phần tử ngẫu nhiên khác);
 - In lần lượt 8 quân bài cho người thứ nhất, thứ hai, thứ ba và thứ tư.

Bài tập tuần 6

Lưu ý về phần bài tập lập trình

- *Tất cả các dữ liệu đầu vào đều được nhập theo cách lấy từ tệp văn bản*
- *Tất cả các chức năng được yêu cầu trong mỗi chương trình phải được viết thành chương trình con riêng (là hàm), hàm `main` chỉ làm việc nhận giá trị từ đối dòng lệnh và gọi tới các chương trình con.*

1. Viết chương trình nhập vào một mảng số nguyên a gồm n phần tử. Biến đổi mảng sao cho các phần tử trong mảng được đổi chỗ cho nhau theo thứ tự ngược lại: phần tử đầu tiên thành phần tử cuối cùng và ngược lại v.v.

2. * Viết chương trình nhập vào một mảng số nguyên a gồm n phần tử. Tìm dãy số nguyên liên tiếp dài nhất xuất hiện trong a thoả mãn các phần tử trong dãy có giá trị bằng nhau. In ra màn hình chiều dài của dãy dài nhất và giá trị của phần tử trong dãy.

Ví dụ: nếu mảng $a = \{1\ 2\ 2\ 1\ 5\ 1\ 1\ 7\ 7\ 7\ 7\ 1\ 1\}$ thì chương trình sẽ in ra màn hình hai số tương ứng là 4 7

3. Viết chương trình `WordCount.java` nhập vào một chuỗi kí tự, đếm và in ra số các từ xuất hiện trong chuỗi.

4. **Hình vuông ma thuật***. Viết chương trình `MagicSquare.java` đọc vào một số nguyên lẻ n và in ra màn hình một *hình vuông ma thuật* có kích thước $n \times n$. Hình vuông ma thuật là hình vuông chứa các số nguyên từ 1 đến n^2 , mỗi số một lần thoả mãn tổng các phần tử trên các hàng, cột và đường chéo là bằng nhau.

4	9	2		11	18	25	2	9
3	5	7		10	12	19	21	3
8	1	6		4	6	13	20	22
				23	5	7	14	16
				17	24	1	8	15

Gợi ý: Sắp xếp các số nguyên từ 1 đến n^2 theo thứ tự tăng dần. Số đầu tiên được gán vào ô có vị trí tại hàng cuối cùng, cột ở giữa. Lặp quá trình gán số nguyên tiếp theo vào ô hàng xóm liền kề theo đường chéo, phía dưới, bên phải. Nếu ô này đã được gán số thì sử dụng ô liền kề phía trên. Sử dụng kỹ thuật "quấn quanh" để xử lý trường hợp biên.

Câu hỏi và bài tập tuần 7

1. Ý nghĩa của phương thức gcdlike() là gì?

```
public static boolean gcdlike(int p, int q) {  
    if (q == 0) return (p == 1);  
    return gcdlike(q, p % q);  
}
```

2. Cho a, b là hai số nguyên dương, phương thức mystery() sau thực hiện việc gì?

```
public static int mystery(int a, int b) {  
    if (b == 0) return 0;  
    if (b % 2 == 0) return mystery(a+a, b/2);  
    return mystery(a+a, b/2) + a;  
}
```

Nếu trong phương thức mystery() ta thay phép toán + thành * và thay return 0 thành return 1 thì ý nghĩa của phương thức mystery() mới thu được là gì?

3. Cho phương thức sau, mystery(0, 8) thực hiện việc gì?

```
public static void mystery(int a, int b) {  
    if (a != b) {  
        int m = (a + b) / 2 ;  
        mystery(a, m);  
        System.out.println(m);  
        mystery(m, b);  
    }  
}
```

Lưu ý về phần bài tập lập trình

-
- Tất cả các dữ liệu đầu vào đều được nhập theo cách lấy từ tệp văn bản
 - Tất cả các chức năng được yêu cầu trong mỗi chương trình phải được viết thành chương trình con riêng (là hàm), hàm `main` chỉ làm việc nhận giá trị từ đối dòng lệnh và gọi tới các chương trình con.

1. Viết chương trình nhập n tính và in ra giá trị $n!$ bằng 2 phương pháp lặp và đệ quy.
2. Viết chương trình nhập n , tính và in ra giá trị số Fibonacci thứ n bằng 2 phương pháp lặp và đệ quy.

Nhắc lại cách tính số Fibonacci thứ n : $f_n = f_{n-1} + f_{n-2}$ với $n \geq 2$; $f_0 = 0, f_1 = 1$.

3. Viết chương trình nhập n và in ra biểu diễn nhị phân của một số nguyên dương n bằng 2 phương pháp lặp và đệ quy.
4. Viết chương trình `Collatz.java` cho phép in ra chuỗi Collatz bắt đầu bằng số nguyên dương n và kết thúc bằng 1 sử dụng phương pháp đệ quy.

Ví dụ: nếu $n = 13$ thì chuỗi Collatz là:

$$13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1.$$

Chú ý: Chuỗi Collatz là một chuỗi số tự nhiên, trong đó số đứng liền sau được xác định theo số đứng liền trước và dừng lại khi gặp phần tử 1. Công thức truy hồi

$$n = \begin{cases} n/2, & \text{nếu } n \text{ chẵn} \\ n \times 3 + 1, & \text{nếu } n \text{ lẻ} \end{cases}$$

Phần tử đầu tiên của chuỗi Collatz được gọi là phần tử sinh. Có giả thiết (chưa được chứng minh hay bác bỏ) là mỗi chuỗi Collatz

luôn kết thúc sau hữu hạn bước và bằng 1. Giả thiết Collatz đã được kiểm chứng đúng đến $n = 2^{60}$.

5. (*) Viết chương trình `Subsequence.java` cho phép in ra tất các các xâu con có chiều dài k của xâu s bằng phương pháp đệ quy.

Ví dụ: Nếu xâu s là `abcd` thì các xâu con có chiều dài $k = 3$ của xâu s là `abc abd acd bcd`.

6. (*) Bài toán tháp Hà Nội: Cho n cái đĩa kích thước khác nhau, có lỗ ở giữa và 3 cái cọc có thể xuyên qua lỗ trên mỗi đĩa. Ban đầu, n đĩa này được xếp theo thứ tự kích thước nằm xuyên trên một cái cọc, đĩa bé nhất nằm trên cùng - tạo thành một dạng hình nón. Viết chương trình mô tả các bước dịch chuyển cần thực hiện để chuyển n đĩa này sang một cái cọc khác, tuân theo các quy tắc sau:

- Mỗi lần chỉ được di chuyển một cái đĩa;
- Không được đặt đĩa lớn hơn lên một đĩa nhỏ hơn.

Bài tập tuần 15

Quản lý sinh viên Mỗi đối tượng sinh viên theo học tại Trường Đại học Khoa học Tự nhiên sẽ có một số thông tin về mã sinh viên (*MSV* - dạng xâu), name (*họ và tên*), birthyear (*năm sinh*), address (*địa chỉ*), GPA (*điểm trung bình*). Bên cạnh việc quản lý hàng nghìn sinh viên đang theo học, mỗi năm Nhà trường sẽ tuyển mới hàng nghìn sinh viên khác. Do đó việc ứng dụng công nghệ thông tin trong quản lý là một trong những yếu tố góp phần nâng cao chất lượng công tác quản lý sinh viên. Ngoài ra để phát huy tốt năng lực của sinh viên trong Trường, Nhà trường đã đặt bài toán xây dựng chương trình quản lý sinh viên cho sinh viên ngành MT& KHTT cũng như ngành TTUD trong khoa Toán - Cơ - Tin học. Yêu cầu cụ thể của chương trình quản lý gồm các chức năng chính sau:

1. Thêm một hay nhiều sinh viên, cập nhật vào tệp nhị phân chứa dữ liệu các đối tượng sinh viên;
2. Sửa thông tin sinh viên có MSV nào đó đã cho, cập nhật dữ liệu vào tệp nhị phân;
3. Tìm kiếm sinh viên theo MSV nào đó đã cho và in kết quả tìm kiếm vào một tệp văn bản;
4. Xóa sinh viên có MSV nào đó đã cho, cập nhật dữ liệu vào tệp nhị phân;
5. Sắp xếp sinh viên theo thứ tự giảm dần của GPA, in kết quả sắp xếp vào một tệp văn bản;
6. Sắp xếp sinh viên theo name, in kết quả sắp xếp vào một tệp văn bản.

Yêu cầu:

- Viết lớp `Student` để quản lý thông tin cho mỗi sinh viên.
- Viết lớp `StudentInfo` để đọc và ghi thông tin đối tượng sinh viên vào tệp

-
- Viết lớp `StudentManager` để cung cấp các phương thức quản lí sinh viên
 - Viết lớp `ManagerTest` để kiểm tra các chức năng của chương trình