String

In a file name (say a.java) type following program. The file can be compiled by command **javac a.java**. To run the program give command **java kapil**.

import java.io.\*;

import java.lang.\*;

class kapil

{ public static void main(String args[])

{ String a,b;

try

{ DataInputStream o=new DataInputStream(System.in);

a=o.readLine();

b=a.substring(2,5);

System.out.println(b);

}

catch(IOException e) { System.out.println(e); }

}

}

The program outputs sub string between positions 2 and 5 (including 2 but excluding 5). The first character is at 0th position. e.g. input qwertyuiuo output ert.

b=a.substring(4);System.out.println(b);

The program outputs string on and after 4th position. input qwertyuiuo output tyuiuo

String a,b,c; a=o.readLine();b=o.readLine();

c=a+b; System.out.println(c);

The program takes two strings and joins them.

1. Write program to remove 2nd letter. Let the input string is pwsxtpbcderxrtxgt then output is pwxtpbcderxrtxgt.
2. Write program to add 0th letter in the beginning. In above case ppwsxtpbcderxrtxgt
3. Write program to exchange first two letters. In above case wpsxtpbcderxrtxgt.
4. Write program to exchange 4th and 10th letter. In above case pwsxrpbcdetxrtxgt.
5. Write program to insert ‘t’ between 1st and 2nd letter. In above case pwtsxtpbcderxrtxgt. [Hint: a+”t”+b].

a=o.readLine(); i=a.indexOf('x'); System.out.println(i);At what location 'x' is present. If more than one occurrence of 'x' is there then the location of first 'x' is returned. If 'x' is absent then 1 is returned. e.g. input wedxtyhxu output 3

int i;String a,b,c,d;i=a.indexOf('x');

b=a.substring(0,i);c=a.substring(i+1);

d=b+c; System.out.println(d);

The first 'x' in the given string is deleted.

1. Write a program, which reads a string and finds string after the first x. Let the input string is pwsxtpbcderxrtxgt then output is tpbcderxrtxgt.
2. Write program to replace first x by y. In above case pwsytpbcderxrtxgt.
3. Write program to output the location of second x. In above case 11.
4. Write program to print the string between 1st and 2nd x. In above case tpbcder.
5. Write program to find string before 2nd x. In above case pwsxtpbcder.
6. Write program to delete the string between 1st and 2nd x. In above case pwsxxrtxgt.
7. Program to exchange the string between 1st and 2nd x, with the string before 1st x. In above case tpbcderxpwsxrtxgt
8. Write program to exchange neighbors of first x. In above case pwtxspbcderxrtxgt.

char b;a=o.readLine();b=a.charAt(2);

System.out.println(b);

Program outputs character at location 2. e.g. input qwertyuiuo output e

1. Write program to find 2nd location of 0th letter. Input pwerpty output 4.
2. Write program, which will delete 1st y immediately after 1st x. If input string is pgyeryuyixaysdyexer then output is pgyeryuyixasdyexer.
3. Write program to exchange neighbors of first occurrence of left neighbors of first ‘x’. e.g. input abcdefxgh output abcdxfegh. input abfcdefxgh output acfbdefxgh.
4. Write program to replace first occurrence of right neighbor of 2nd x by left neighbor of 1st x. Input imgpxugxutkl output imgpxpgxutkl. Input bcxdefxgh output bcxdefxch.
5. Write program, which reads a string. Let x and y be respectively left and right neighbors of the second occurrence of the 0th letter. Find the substring between first occurrence of y and (first occurrence of x after first occurrence of y). e.g. input patkgfmpkst output kgfm. Input pastgksfsptse output tgks. Input raklfrgmcfd output gmcf. Input ywetyykjhtl output ywet.

a=o.readLine();b=o.readLine();

i=a.compareTo(b);System.out.println(i);

Input two strings. Output is 0 if both are same. If second string is (lexicographically) bigger then a negative number is outputted. If first string is bigger then some positive number is outputted.

Use of “if” is permitted in following programs.

1. Read two strings. Print lexicographically bigger string first and smaller later.
2. Read two string. Print 1 if first string is bigger, 2 if second string is bigger, 0 if both are same.
3. Read three strings. Print 1 if first string is biggest, 2 if second string is biggest, 3 if 3rd string is biggest, 0 if all are same, -1 if 1st and 2nd string are biggest, -2 if 2nd and 3rd string are biggest, -3 if 1st and 3rd string are biggest.

a=o.readLine();a=a.trim();

i=a.indexOf(' ');b=a.substring(0,i);

System.out.println(b);

Print first word. Trim removes blank spaces at the beginning and at the end. If it is not used then the problem will arise if blanks are given at the beginning. The program will not work if string has only one word. If string is ram Prasad dey then output is ram.

1. Print second word. In above case Prasad.
2. Delete second word. In above case ram dey.
3. Exchange first and second word. In above case Prasad ram dey.
4. Exchange first letters of first two words. In above case Pam rrasad dey.
5. Exchange last letters of first two words. In above case rad Prasam dey.
6. Find the location of first ‘a’ in second word. In above case 6.
7. Find location of first letter of first word in second word. In above case 5.

String Buffer

import java.io.\*;

import java.lang.\*;

class kapil

{ public static void main( String args[])

{ StringBuffer a;String b;int i;

try

{ DataInputStream o=new DataInputStream(System.in);

a=new StringBuffer("");b=o.readLine();a.append(b);

System.out.println("The input string is"+a);

i=a.indexOf("x");System.out.println(i);

i=a.indexOf("xy");System.out.println(i);

a.setCharAt(6,'X');System.out.println(a);

a.insert(4,"tty");System.out.println(a);

a.append("ghij");System.out.println(a);

a.reverse();System.out.println(a);

a.delete(4,7);System.out.println(a);

a.deleteCharAt(5);System.out.println(a);

a.replace(3,7,"was");System.out.println(a);

a.setLength(5);System.out.println(a);

b=a.toString();System.out.println(b);

}

catch(Exception e)

{ System.out.println("Error"+e); }

}

}

All the programs should be done without using +(plus), (minus) or append (except during converting input string to string buffer).

1. Read a string and replace first x by y. Input artxuxtxxp output artyuxtxxp.
2. Replace last x by y. (do not use loop)
3. Replace second x by y.
4. Replace all x’s by y.
5. Find the number of x’s. In above case 4.
6. Whether given string is a Palindrome. A Palindrome is word which remains same when reversed. e.g. malayalam.
7. Find length of the string. (Do not use *length( )*)
8. Suppose string has words. e.g. Ram is a good boy. Find first and second word.
9. Find the first word of 4 letters. In above case *good*. Assume such word present at least once.
10. Remove all blanks. In above case *Ramisagoodboy*.
11. Find last letter of first word.
12. Find first letter of last word.
13. Find first letter of every word.
14. Replace first letter of every word by x. In above case x*am xs x xood xoy*.
15. Word wise reverse of the string. In above case *boy good a is Ram*.
16. Reverse each word. In above case output is *maR si a doog yob*.
17. Interchange first two words. In above case *is Ram a good boy*.
18. Write program, which reads an expression and outputs its value. Assume that parenthesis is not used. Assume all operations are left associative and all numbers are integers. Take only ‘+’ and ‘\*’ as operators. e.g. input 12+17\*10 output 290.