

SIKSHA 'O' ANUSANDHAN
DEEMED TO BE UNIVERSITY

Admission Batch:

Session:

Laboratory Record

Computer Science Workshop 2 (CSE 3141)

Submitted by

Name: Saswat Mohanty

Registration No.: 1941012407

Branch: Computer Science and Engineering (CSE)

Semester: 4th Section: D



Department of Computer Science & Engineering

Faculty of Engineering & Technology (ITER)

Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha - 751030

INDEX

Assignment on String

A1) Write a program to convert a string to integer and vice-versa.

Program :-

```
import java.util.*;  
public class A3Q1 {  
    public static int stringToInt (String s) {  
        int length = s.length ();  
        int sign = 1;  
        int f_index = 0;  
        int res = 0;  
        if (f_index == length)  
            return 0;  
        if (s.charAt (f_index) == '-' || s.charAt (f_index) == '+') {  
            if (s.charAt (f_index) == '-')  
                sign = -1;  
            f_index++;  
        }  
        for (int i = f_index; i < length; i++) {  
            char c = s.charAt (i);  
            if (Character.isDigit (c)) {  
                int num = c - '0';  
                res = (res * 10) + num;  
            }  
            else  
                break;  
        }  
        return res * sign;  
    }  
}
```

```
public static intToString(int n) {
    boolean checkneg = n < 0;
    StringBuilder sb = new StringBuilder();
    n = checkneg ? -n : n;
    List<Integer> list = new ArrayList<Integer>();
    int digitleft = n;
    int currentdigit = 0;
    while (true) {
        currentdigit = digitleft % 10;
        list.add(currentdigit);
        digitleft /= 10;
        if (digitleft == 0)
            break;
    }
    currentdigit = list.remove(0);
    sb.append(currentdigit);
    for (int c : list)
        sb.append(c);
    sb.reverse().insert(0, checkneg ? '-' : '');
    return sb.toString().trim();
}

public static void main (String [] args) {
    System.out.println (((Object) stringToInt ("-2234")).getClass().
        getSimpleName ());
    System.out.println (((Object) intToString (-2234)).getClass().
        getSimpleName ());
    System.out.println (stringToInt ("-2234"));
    System.out.println (intToString (-2234));
}
```

Output :-

Integer

String

-2 2 3 4

-2 2 3 4

Q2) Write a program that performs base conversion. The input is a string, an integer b_1 , and another integer b_2 . The string represents the integer in base b_1 . The output should be the string representing the integer in base b_2 .

Program :-

```
import java.util.*;
public class A3Q2 {
    public static String base(String str, int base1, int base2) {
        int end = 0;
        if (str.charAt(0) == '-') {
            end = 1;
        }
        int num = 0, c = 0;
        char ch;
        for (int i = str.length() - 1; i >= end; i--) {
            ch = str.charAt(i);
            if (Character.isDigit(ch)) {
                num = num + (((int)(Math.pow(base1, c))) * (ch - '0'));
            } else {
                num = num + (((int)(Math.pow(base1, c))) * (ch - 'A' + 10));
            }
            c = c + 1;
        }
        if (end == 1)
            num = -num;
        if (num == 0)
            return "0";
        int re;
        String res = "";
        int temp = num;
        while (temp != 0) {
            re = temp % base2;
            res = res + Character.toString(re);
            temp = temp / base2;
        }
        return res;
    }
}
```

Name: Saswat Mohanty

36

Regd. Number: 1941012407

```
if (num < 0)
    tmph = -tmph;
while (tmph != 0) {
    r = tmph % base2;
    if (r < 10)
        res = res + res;
    else
        res = (char) ('A' + (r - 10)) + res;
    tmph = tmph / base2;
}
if (num < 0)
    return ("-" + res);
return (res);
}

public static void main (String [] args) {
Scanner sc = new Scanner (System. in);
System.out.println ("Enter string");
String str = sc.nextLine();
System.out.println ("Enter the Base 1 Value");
int base1 = sc.nextInt();
System.out.println ("Enter the Base 2 Value");
int base2 = sc.nextInt();
System.out.println (base (str, base1, base2));
}
```

Output:-

```
Enter string
esffebdvyew
Enter the Base1 Value
2
```

Enter the Base & Value

2

11000111001111.000

Q3) Implement a function that converts a spreadsheet column id to the corresponding integer with "A" corresponding to 1. For example, you should return 4 for "D", 27 for "AA", 702 for "ZZ", etc. How would you test your code.

Program :-

```
import java.util.*;  
public class A3Q3 {  
    public static int column(String s) {  
        int colcount = 0, i = 0, len = s.length();  
        while (len > 0) {  
            colcount *= 26;  
            colcount += s.charAt(i) - 'A' + 1;  
            i++;  
            len--;  
        }  
        return colcount;  
    }  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the string");  
        String s = sc.next();  
        System.out.println(column(s.toUpperCase()));  
    }  
}
```

Output :-

Enter the string
Lorem ipsum dolor sit amet
5759663

- Q4) Write a program which takes as input an array of characters and removes each 'b' and replaces each 'a' by two 'd's. Specifically along with the array, you are provided an integer-valued size. size denotes the number of entries of the array that the operation is to be applied to. You do not have to worry preserving about subsequent entries. For example, if the array is [a, b, a, c] and size is 4, then you can return [d, d, d, d, c]. You can assume there is enough space in the array to hold the final result.

Program :-

```
import java.util.*;  
public class A3Q4 {  
    public static int removeElement (char [] arr) {  
        int length = arr.length, c = 0, a = 0;  
        for (int i = 0; i < length; i++) {  
            if (arr[i] != 'b')  
                arr[c++] = arr[i];  
            if (arr[i] == 'a')  
                a++;  
        }  
        int ch = c - 1;  
        c = a - 1;  
        int size = c + 1;  
        while (ch > -1) {  
            if (arr[ch] == 'a') {  
                arr[c--] = 'd';  
                arr[c--] = 'd';  
            }  
        }  
    }  
}
```

Name: Sarwat Nahanty

39

Regd. Number: 1941012407

```
        else
            arr[c--] = arr[cp];
        }
    return size;
}

public static void main (String [] args) {
    char [] arr = { 'a', 'c', 'd', 'b', 'b', 'b', 'c' };
    System.out.println ("Before Removal and Replacing");
    for (char ch: arr) {
        System.out.print (ch + " ");
    }
    System.out.println ("n Size " + removeElement (arr));
    System.out.println ("After Removal and Replacing");
    for (char ch: arr) {
        System.out.print (ch + " ");
    }
}
```

Output:-

Before Removal and Replacing

a c d b b b c

Size 5

After Removal and Replacing

d d c d c b b c

Q5) Implement a function which takes as input a string s and returns true if s is a palindromic string.

Program :-

```
import java.util.*;
public class A3Q5 {
    public static boolean isPalindrome (String s) {
        String ss = s.toLowerCase().replaceAll ("[^a-zA-Z]", " ");
        return ss.equals (new StringBuilder (ss).reverse ().toString ());
    }
    public static void main (String [] args) {
        System.out.println (isPalindrome ("Able was I, ere I saw
            Elba !"));
    }
}
```

Output :-

true.

- Q6) Write a program which takes as input a phone number, specified as a string of digits, and returns all possible character sequences that correspond to the phone number. The cell phone keypad is specified by a mapping that takes a digit and returns the corresponding set of characters. The character sequences do not have to be legal words or phrases.

Program :-

```
import java.util.*;
public class A3Q6 {
    static String [] keycode = {"", "", "abc", "def", "ghi", "jkl", "mno",
        "pqrs", "tuv", "wxyz"};
    public static List <String> combinations (String numinp) {
        if (numinp.length () == 0) {
            return new ArrayList <String> ();
        }
    }
}
```

Name: Sarwat Mohanty

41

Regd. Number: 1941012407

```
        }  
        if (numinp.length() == 1) {  
            int index = numinp.charAt(0) - '0';  
            String code = keycode[index];  
            List<String> l = new ArrayList<>(c);  
            for (char ch : code.toCharArray()) {  
                l.add(" " + ch);  
            }  
            return l;  
        }  
        char ch = numinp.charAt(0);  
        String str = numinp.substring(1);  
        List<String> l = combinations(str);  
        int index = ch - '0';  
        String code = keycode[index];  
        List<String> res = new ArrayList<>(c);  
        for (int i = 0; i < code.length(); i++) {  
            char ch2 = code.charAt(i);  
            for (String ch3 : l) {  
                res.add(" " + ch2 + ch3);  
            }  
        }  
        return res;  
    }  
    public static void main(String[] args) {  
        System.out.println(combinations("234"));  
    }  
}
```

Output :-

[adg, adh, adi, aeg, aeh, ael, afg, afh, afi, bdg, bdh, bdl, beg, beh, bel, bfg, bfh, bfj, cdg, cdh, cdj, ceg, ch, ci, cfi, cgh, cjh, cfj]

Q7) Write a program that takes as input an integer n and returns the n^{th} integer in the look-and-say sequence.
Return the result as a string.

Program :-

```
import java.util.*;  
public class A3Q7{  
    public static String lookAndSay(int n){  
        String s = "1";  
        for(int i=1; i<n; i++)  
            s = nextNumber(s);  
        return s;  
    }  
  
    private static String nextNumber(String s){  
        StringBuilder result = new StringBuilder();  
        for(int i=0; i<s.length(); i++){  
            int count = 1;  
            while(i+1 < s.length() && s.charAt(i) == s.charAt(i+1))  
                i++;  
            count++;  
        }  
        result.append(count);  
        result.append(s.charAt(i));  
    }  
}
```

```
        return result.toString();
    }

public static void main (String [] args) {
    System.out.println (lookAndSay (2));
}
```

Output :-

11

Q3) Write a program which takes as input a valid Roman Number string s and returns the integer it corresponds to.

Program :-

```
import java.util.*;
public class A3Q8 {
    static int toInt (String s) {
        Map<Character, Integer> m = new HashMap<Character,
                                            Integer>();
        put ('I', 1);
        put ('V', 5);
        put ('X', 10);
        put ('L', 50);
        put ('C', 100);
        put ('D', 500);
        put ('M', 1000); }

        int res = 0;
        int n = s.length();
        for (int i=0; i<n; i++) {
            if (i!=n-1 && m.get(s.charAt(i)) < m.get(s.charAt(i+1))) {
```

```
res += m.get(s.charAt(i+1)) - m.get(s.charAt(i));  
i++;  
}  
else  
    res += m.get(s.charAt(i));  
}  
return res;  
}  
  
public static void main (String [] args) {  
    System.out.println (print ("LIX"));  
}
```

Output:-

59

Q9) Write a program that determines where to add periods to a decimal string so that the resulting string is a valid IP address. There may be more than one valid IP address corresponding to a string, in which case you should print all possibilities. For example, if the mangled string is "19216811" then two corresponding IP addresses are 192.168.1.1 and 19.216.81.1 (There are seven other possible IP addresses for this string.)

Program:-

```
import java.util.*;  
public class A3Q9 {  
    public static List<String> validip (String s) {  
        List<String> res = new ArrayList<String>();  
        addchar (s, 0, new ArrayList<String>(), res);  
        return res;  
    }
}
```

Name: Sirwat Mohsin

45

Regd. Number: 1941012407

```
public static void addchar(string s, int n, List<string>  
    frecom, List<string> res) {  
    if (n == s.length() && frecom.size() == 4) {  
        res.add(build(frecom));  
        return;  
    }  
    for (int j = n + 1; j <= s.length() && j - n <= 3; j++) {  
        string sub = s.substring(n, j);  
        if (isValidPart(sub) && frecom.size() <= 3) {  
            frecom.add(sub);  
            addchar(s, j, frecom, res);  
            frecom.remove(sub);  
        }  
    }  
}  
private static string build(List<string> partList) {  
    stringBuffer sb = new StringBuffer();  
    for (int i = 0; i < 4; i++) {  
        sb.append(partList.get(i));  
        if (i != 3)  
            sb.append(",");  
    }  
    return sb.toString();  
}  
private static boolean isValidPart (string s) {  
    if (s.length () > 3)  
        return false;  
}
```

```
if (s.startsWith("0") && s.length() > 0)
    return false;
int subval = Integer.parseInt(s);
return subval <= 255 && subval >= 0;
```

{

```
public static void main (String [] args) {
    System.out.println (isValid ("19216811"));
}
```

}

Output :-

```
[192.168.1.11, 19.2.168.11, 19.21.68.11, 19.216.8.11,
19.216.81.1, 192.68.1.11, 192.16.8.11, 192.16.81.1,
192.168.1.1]
```

Q10) Write a program which takes as input a string s and returns the snakestring of s.

Program :-

```
import java.util.*;
public class A3Q10 {
    public static void main (char [][] arr, String s) {
        int row = 1;
        boolean flag = true;
        for (int i = 0; i < arr.length; i++) {
            arr [row] [i] = s.charAt (i);
            if (row == 3)
                flag = false;
            else if (row == 1)
                flag = true;
```

```
if(flag)
    row++;
else
    row--;
}
for (int i = 1; i < 4; i++) {
    for (int j = 0; j < arr.length; j++) {
        System.out.print(arr[i][j] + " ");
    }
    System.out.println();
}
}

public static void main (String [] args) {
    String s = "HelloWorld";
    int n = s.length();
    char [][] arr = new char [n] [n];
    sinwave (arr, b);
}
}
```

Output:-

H
e l o
l l W r d
o

Q11) Run-length encoding (RLE) compression offers a fast way to do efficient on-the-fly compression and decompression of strings. The idea is simple encode successive repeated characters by the repetition count and the character. For example, the RLE of "aaaabccaa" is "3a2b3c2a". The decoding of "3e4f2e" returns "eeeffffeee". Implement run-length encoding and decoding functions. Assume the string to be encoded consists of letters.

of the alphabet, with no digits, and the string to be decoded
is a valid encoding.

Program :-

```
import java.util.*;
public class A3Q11 {
    public static String decoding (String s) {
        int c = 0;
        StringBuilder res = new StringBuilder();
        for (int i = 0; i < s.length(); i++) {
            char ch = s.charAt(i);
            if (Character.isDigit(ch))
                c = c * 10 + ch - '0';
            else {
                while (c > 0)
                    res.append(ch);
                c--;
            }
        }
        return res.toString();
    }

    public static String encoding (String s) {
        int c = 1;
        StringBuilder res = new StringBuilder();
        for (int i = 1; i <= s.length(); i++) {
            if (i == s.length() || s.charAt(i) != s.charAt(i - 1)) {
                res.append(c);
                res.append(s.charAt(i - 1));
                c = 1;
            }
        }
        return res.toString();
    }
}
```

```
        else
            c++;
    }
    return res.toString();
}

public static void main (String [] args) {
    System.out.println (decoding ("3e4f2c"));
    System.out.println (encoding ("aaaabccca"));
}
}
```

Output:-

cefffffe

a1b3c2a

Q12) Given two strings s (the "search string") and t (the "text"), find the first occurrence of s in t.

Program :-

```
import java.util.*;
public class A3Q12 {
    public static int rabinKarp (String t, String s) {
        if (s.length () > t.length ())
            return -1;
        final int BASE = 26;
        int tHash = 0, sHash = 0;
        int powerS = 1;
        for (int i=0; i<s.length (); i++) {
            powerS = i > 0 ? powerS * BASE : 1;
```

```
tHash = tHash * BASE + t.charAt(i);  
sHash = sHash * BASE + s.charAt(i);  
}  
for (int i = s.length(); i < t.length(); i++) {  
    if (tHash == sHash && t.substring(i - s.length(), i).  
        equals(s))  
        return i - s.length();  
    tHash = t.charAt(i - s.length()) * powers;  
    tHash = tHash * BASE + t.charAt(i);  
}  
if (tHash == sHash && t.substring(t.length() - s.length()).equals(s))  
    return t.length() - s.length();  
return -1;  
}  
public static void main (String [] args) {  
    System.out.println (KarpRabin ("GACGCGA", "CGC"));  
}
```

Output :-

2