Main class **Source** should have the functionality to validate the input hexadecimal and decimal colour codes.

**Create two static methods in class ColourCodeValidator as per the below signature**

validateHexCode(String):**int**

validateDecimalCode(String):**int**

Both the methods **return** 1 **for** valid codes **and** -1 **for** invalid codes. Rules **for** valid codes are **given** below

**Hexadecimal code rules**

* Format: #A1BC23
* Must start with "#" symbol
* Must contain six characters after #
* It may contain alphabets from A-F or digits from 0-9

**Decimal code rules**

* Format: rgb(x,y,z)
* x,y and z are values from 0 to 255 inclusive

**In the main method , do the following**

* Accept the inputs using Console as shown in the Example section
* First input is choice based on which one of the static methods should be invoked
* choice 1 is for validating the input hexadecimal colour code
* choice 2 is for validating the input decimal colour code
* Display **Valid code** or **Invalid code** based on the validation result
* If the choice is neither 1 or 2, display message "Invalid choice"

**Example**

Sample Input:

1 #ABCDEF

Expected Output:

Valid Code

Sample Input:

2 rgb(9,99,249)

Expected Output:

Valid Code

Sample Input:

9

Expected Output:

Invalid choice

Solution:

import java.util.Scanner;

class ColourCodeValidator

{

}

public class Source{

public static boolean isNumeric( String str) {

// null or empty

if (str == null || str.length() == 0) {

return false;

}

for (char c : str.toCharArray()) {

if (!Character.isDigit(c)) {

return false;

}

}

return true;

}

public static int validateHexCode(String str)

{ int flag=0;

char[] ar= {'A','B','C','D','E','F','0','1','2','3','4','5','6','7','8','9'};

if(str.charAt(0)=='#'&&str.length()==7)

{ for(int i=1;i<str.length();i++){

for(int j=0;j<ar.length;j++){

if(str.charAt(i)==ar[j])

{

flag++;

}

}

}

}else

{

flag++;

}

if(flag==6){

return 1;

}

else {

return -1;

}

}

public static int validateDecimalCode(String str)

{ int flag=0;

String s=str.substring(0,3);

if(s.equals("rgb")){

String s1= str.substring(4,str.length()-1);

String[] arr=s1.split(",");

for(int i=0;i<arr.length;i++){

String s11=arr[i];

if(isNumeric(s11)==true) {

int co=Integer.parseInt(s11);

for(int j=0;j<256;j++){

if(co==j){

flag++;

}

}

}

else {

return -1;

}

}

}

if(flag==3){

return 1;

}

else {

return -1;

}

}

public static void main(String args[]) throws Exception{

Scanner sc = new Scanner(System.in);

int i=sc.nextInt();

String s=sc.next();

ColourCodeValidator color= new ColourCodeValidator();

if(i==1&&Source.validateHexCode(s)==1){

System.out.println("Valid code");

}

else if(i==2&&Source.validateDecimalCode(s)==1){

System.out.println("Valid code");

}

else if(i==1||i==2)

{

System.out.println("Invalid code");

}

else{

System.out.println("Invalid choice");

}

}

}

Harry has recently learned about strings in his programming classes. He decided to create some interesting strings using the basic concepts.

Help Harry!

Your task here is to implement a **JAVA** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider **default visibility** of classes, data fields and methods unless mentioned otherwise. All the methods that you are implementing should be non-static.

**Specification:**

**class** **definitions**:

**class** **StringPlay**:

  data fields:

**int** convert

**int** max;

StringPlay(): Define an empty constructor with **public** visibility.

**class** **StringMethods**:

   convertToInt(StringPlay sp, String str):

    visibility: **public**

**return** type:**int**

getMax(StringPlay sp, String str, **char** ch):﻿

visibility: **public**

**return** type:**int**

**Task:**

class **StringPlay**

- Implement **StringPlay**class according to the above specifications

**class StringMethods**

- Implement the below methods for this class:

* **int convertToInt(StringPlay sp, String str):**Convert the string str to int, return the int value and assign the value to suitable sp variable(convert). All the strings will contain only numbers.
* Example: str = "123" then resultant is 123.
* **int getMax(StringPlay sp, String str, char ch):**Return the total number of char ch present in string str and assign the value to sp variable max and return the same value.
* Example: str = "This is good" , ch = 'o' then resultant value = 2
* Example: str = "doselect Et le", ch='e' then resultant value = 3

**Sample Input**

StringMethods sm = **new** StringMethods();

StringPlay sp = **new** StringPlay();

sm.getMax(sp,"fgfgfgf",'g')

sm.convertToInt(sp,"123")

**Sample Output**

3

123

class StringPlay{

//Write Your Code Here..

int convert;

int max;

public StringPlay(){

this.convert=convert;

this.max=max;

}

}

class StringMethods{

//Write Your Code Here..

public int convertToInt(StringPlay sp,String str){

int n=Integer.parseInt(str);

sp.convert=n;

return n;

}

public static int getMax(StringPlay sp, String str,char ch){

int flag=0;

for(int i=0;i<str.length();i++){

if(str.charAt(i)==ch)

flag++;

}

sp.max=flag;

return flag;

}

}

public class Source {

public static void main(String args[] ) throws Exception {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

StringMethods sm=new StringMethods();

StringPlay sp=new StringPlay();

sm.getMax(sp,"fsfff",'g');

sm.convertToInt(sp,"123");

System.out.println(sp.convert);

System.out.println(sp.max);

}

}

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

### Student Info Extraction

* subject Coding
* casino 100 points

**DESCRIPTION**

Your task here is to implement a **JAVA** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider **default visibility** of classes, data fields and methods unless mentioned otherwise.

**Specifications:**

class definitions:

  class Student:

      data fields:

        name: String variable

        usn: String Variable

        college: String variable

cgpa : int variable

      Constructor to initialize the class variables.

public Student(name,usn,college,cgpa)

      ﻿

  class StudentImplementation:

      method definitons:

        Student getStudentInfo(String str): To extract the name,usn,college and cgpa from the String str and return a student object.

            return type: Student object

            visibility: public

        String getStudentSection(Student s): **Use** the usn **of** the Student s **and** **return** the **section** he/she **is** in. **If** **last** three digit **of** student usn **is** **between** 001-060 **return** "A", **if** usn **is** **between** 061-120 **return** "B" , **if** usn **is** **between** 121-180 **return** "C" **and** **if** usn greater **than** 180 **return** "Z".

**return** **type**: **String**

            visibility: **public**﻿

String manipulation is a tough task and your company wants you to do some string operations and manipulation.

Suppose you are given a string (example **Amit Rai@1PC16CS046-ALU#8):**

* The part before **'@'**represents the name of the student.
* The part before**'-'**and after '**@'**represents the usn of the student.
* The part before **'#'**and after **'-'** represents the college of the student.
* The part after **'#'**represents the cgpa of the student.

**Tasks:**

* Implement the **Student** class according to the specification given.
* Implement the **StudentImplementation**class where you have to implement the following two methods according to the specifications given:

1. **Student getStudentInfo(String str)**
2. **String getStudentSection(Student s)**

**Method Descriptions:**

1. **Student getStudentInfo(String str):**

* takes a String parameter (e.g. **Amit Rai@1PC16CS046-ALU#8**).
* Extract the information from the String str
* Create and populate a Student object with the extracted information and return the Student object

2. **String getStudentSection(Student s):**

* takes a Student object as the parameter.
* Use the last three digits of usn of the Student s and return the section he/she is in.
* If last three digit of student's usn is between 001-060 return "A", if usn is between 061-120 return "B" , if usn is between 121-180 return "C" and if usn is greater than 180 return "Z".

class Student{

String name;

String usn;

String college;

int cgpa;

public Student(String name,String usn,String college,int cgpa){

this.name=name;

this.usn=usn;

this.college=college;

this.cgpa=cgpa;

}

}

class StudentImplementation{

public Student getStudentInfo(String str){

int n=0,u=0,c=0,score=0;

String name="",usn="",college="";

char cgpa;

for(int i=0;i<str.length();i++)

{

char ch=str.charAt(i);

if(ch=='@'){

n=i;

name=str.substring(0,n);

}

if(ch=='-'){

u=i;

usn=str.substring(n+1,u);

}

if(ch=='#'){

c=i;

college=str.substring(u+1,c);

}

}

cgpa=str.charAt(str.length()-1);

score=Integer.parseInt(String.valueOf(cgpa));

Student st=new Student(name,usn,college,score);

return st;

}

public String getStudentSection(Student s){

String roll=s.usn;

int n=roll.length();

String us=roll.substring(n-3,n);

int usn=Integer.parseInt(us);

if(usn>001 && usn<=060){

return "A";

}

if(usn>=061 && usn<=120){

return "B";

}

if(usn>=121 && usn<=180){

return "C";

}

if(usn>=181){

return "Z";

}

return "";

}

}

class Source{

public static void main(String[] args) {

StudentImplementation sti=new StudentImplementation();

Student st=new Student("amit","1PC16CS119","MIT",9);

st=sti.getStudentInfo("Amit Rai@1PC16S046-ALU#8");

System.out.println(st.name+"\n"+st.usn+"\n"+st.college+"\n"+st.cgpa);

System.out.println(sti.getStudentSection(st));

}

}