**/\* Alternate Add and Sub\*/**

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int AddSub(int input1,int input2)

{

//Read only region end

//Write code here

int sum1=0,sum2=0,result1=0;

int[]intarr=new int[input1];

int first=input1;

for(int i=0;i<input1;i++){

intarr[i]=first;

first--;

}

if(input2==1){

for(int i=1;i<input1;i=i+2)

{

sum1+=intarr[i];

}

for(int i=2;i<input1;i=i+2){

sum2+=intarr[i];

}

result1=input1+(sum2-sum1);

}

else if(input2==2){

for(int i=1;i<input1;i=i+2)

{

sum1+=intarr[i];

}

for(int i=2;i<input1;i=i+2){

sum2+=intarr[i];

}

result1=input1+(sum1-sum2);

}

return result1;

}

}

**/\* cyclic sum\*/**

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int sumOfSumsOfDigits(int input1)

{

//Read only region end

int[]a=new int[input1.ToString().Length];

int i=0;

int p=1;

int sum=0;

while(input1>0){

a[i]=input1%10;

input1/=10;

i++;

}

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

sum+=a[j]\*p;

p++;

}

return sum;

}

}

/\* Image Viewer\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int findNumberOfImages(int[] input1,char[] input2)

{

//Read only region end

//Write code here

int m=input1[0];

int n=input1[1];

int o=input1[2];

int p=input1[3];

int sum=0,count=0;

if(input2[0]=='h' && o>p){

count=0;

}

else{

for(int i=0;i<m;i++){

if(i==0){

if(input2[i]=='v'){

sum+=1;

}

if(input2[i]=='h'){

sum+=1+o;

}

}

else{

if(input2[i]=='v'){

sum=1+n+sum;

}

if(input2[i]=='h'){

sum=1+n+o+sum;

}

}

if(sum<=p){

count++;

}

else{

break;

}

}

}

return count;

}

}

/\*roman to string\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public string FindMessage(string input1)

{

//Read only region end

//Write code here

string result = "";

string[] str = input1.Split(' ');

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

{

result += (char)(getInt(str[i]));

}

return result;

}

static int getInt(string n)

{

int s = 0;

if (n.StartsWith("C"))

{

if (n.Length > 1)

s += 100 + getInt(n.Substring(1));

else

s += 100;

}

else if (n.StartsWith("XC"))

{

if (n.Length > 2)

s += 90 + getInt(n.Substring(2));

else

s += 90;

}

else if (n.StartsWith("LXXX"))

{

if (n.Length > 4)

s += 80 + getInt(n.Substring(4));

else

s += 80;

}

else if (n.StartsWith("LXX"))

{

if (n.Length > 3)

s += 70 + getInt(n.Substring(3));

else

s += 70;

}

else if (n.StartsWith("LX"))

{

if (n.Length > 2)

s += 60 + getInt(n.Substring(2));

else

s += 60;

}

else if (n.StartsWith("L"))

{

if (n.Length > 1)

s += 50 + getInt(n.Substring(1));

else

s += 50;

}

else if (n.StartsWith("XL"))

{

if (n.Length > 2)

s += 40 + getInt(n.Substring(2));

else

s += 40;

}

else if (n.StartsWith("XXX"))

{

if (n.Length > 3)

s += 30 + getInt(n.Substring(3));

else

s += 30;

}

else if (n.StartsWith("XX"))

{

if (n.Length > 2)

s += 20 + getInt(n.Substring(2));

else

s += 20;

}

else if (n.StartsWith("X"))

{

if (n.Length > 1)

s += 10 + getInt(n.Substring(1));

else

s += 10;

}

else

{

switch (n)

{

case "I":

s += 1;

break;

case "II":

s += 2;

break;

case "III":

s += 3;

break;

case "IV":

s += 4;

break;

case "V":

s += 5;

break;

case "VI":

s += 6;

break;

case "VII":

s += 7;

break;

case "VIII":

s += 8;

break;

case "IX":

s += 9;

break;

case "X":

s += 10;

break;

default:

break;

}

}

return s;

}

}

/\*generate series and find nth element\*/

#include<stdio.h>

#include<string.h>

// Read only region start

int seriesN(int input1,int input2,int input3,int input4)

{

// Read only region end

// Write code here

int n=input1;

int n1=input2-input1;

int n2=input3-input2;

for(int i=1;i<input4;i++){

if(i%2==0){

n=n+n2;

}

else{

n=n+n1;

}

}

return n;

}

/\*long seq of odd\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int getLen\_LS\_ODD(int[] input1,int input2)

{

//Read only region end

//Write code here

int long\_seq=0,no\_of\_seq=0,c=1;

for(int i=1;i<input2;i++){

if((input1[i]%2)!=0){

c++;

}

else{

if(c>long\_seq && c!=1){

long\_seq=c;

}

if(c>1){

no\_of\_seq++;

}

c=1;

}

}

if(c>long\_seq && c!=1){

long\_seq=c;

}

if(c>1){

no\_of\_seq++;

}

return long\_seq-1;

}

}

/\*statistics of a string\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public string StringStatistics(string input1)

{

//Read only region end

int num\_count=0,sum\_num=0;

int alp\_count=0,j=0;

string z="ZERO";

string s="";

for(int i=0;i<input1.Length;i++){

if(char.IsDigit(input1[i])){

num\_count++;

int a=int.Parse(input1[i].ToString());

sum\_num+=a;

}

else if(char.IsLetter(input1[i])){

alp\_count++;

s+=input1[i];

}

}

Console.WriteLine("num\_count:{0}",num\_count);

Console.WriteLine("alp\_count:{0}",alp\_count);

Console.WriteLine("sum\_num:{0}",sum\_num);

if(num\_count==0 || alp\_count==0 || sum\_num==0){

return z;

}

Console.WriteLine("string:{0}",s);

string result=alp\_count.ToString()+s+sum\_num.ToString();

Console.WriteLine("result:{0}",result);

return result;

}

}

/\* sum of long seq\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lb

{

class Program

{

static void Main(string[] args)

{

int[] a = { 1, 2, 4, 6, 3, 5, 2, 4, 6, 8, 9, 11, 13, 14, 16, 18, 20, 21, 28, 30, 32, 34 };

int count = 0, max = 0, p = 0, c\_index = 0, sum = 0;

int input2 = 22;

for (int i = 0; i < input2; i++)

{

if (a[i] % 2 == 0)

{

count++;

}

else

{

if (count > max)

{

max = count;

}

count = 0;

}

}

if (count > max)

{

max = count;

}

/\* 1, 2, 4, 6, 3, 5, 2, 4, 6, 8, 9, 11, 13, 14, 16, 18, 20, 21, 28, 30, 32, 34 \*/

for (int j = 0; j < input2; j++)

{

if (a[j] % 2 == 0)

{

p++;

c\_index = j;

}

else

{

if (p == max)

{

for (int k = c\_index; k >= c\_index - p + 1; k--)

{

sum += a[k];

}

}

p = 0;

}

}

if (p == max)

{

for (int k = c\_index; k >= c\_index - p + 1; k--)

{

sum += a[k];

}

}

Console.WriteLine(sum);

Console.Read();

}

}

}

/\* token sequence\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public string findSequence(int input1,string[] input2,int[] input3)

{

//Read only region end

//Write code here

string str="";

int[] arr=new int[input1];

for(int i=0;i<input1;i++){

arr[i]=input3[i];

}

Array.Sort(arr);

int cindex=0,maxIndex=0,maxlen=0,count=1;

for(int j=0;j<input1-1;j++){

if((arr[j])+1==arr[j+1]){

count++;

if(count==2){

cindex=j;

}

}

else{

if(count>maxlen && count!=1){

maxIndex=cindex;

maxlen=count;

}

count=1;

}

}

if(count>maxlen && count!=1){

maxIndex=cindex;

maxlen=count;

}

if(maxlen<3){

str="NONE"+":";

}

else{

for(int k=maxIndex;k<(maxIndex+maxlen);k++){

for(int g=0;g<input1;g++){

if(arr[k]==input3[g]){

str+=input2[g]+":";

}

}

}

}

return str.Substring(0,str.Length-1);

}

}

/\*water the flowers\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int FindMinDays(int input1,string input2)

{

//Read only region end

//Write code here

string[]arr=input2.Split(' ');

int[] intarr=new int[arr.Length];

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

{

intarr[i]=int.Parse(arr[i]);

}

Array.Sort(intarr);

Array.Reverse(intarr);

Console.WriteLine(intarr);

int count=0,sum=0;

for(int i=0;i<intarr.Length;i++)

{

sum=sum+intarr[i];

if(sum<input1)

{

count++;

}

else if(sum >= input1)

{

count++;

break;

}

}

Console.Write(count);

if(input1 == 0)

return 0;

else if(sum < input1)

return -1;

else

return count;

}

}

/\* weight of a string\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int weightOfString(string input1,int input2)

{

//Read only region end

//Write code here

input1=input1.ToUpper();

string s=input1;

int result=0;

if(input2==1){

for(int i=0;i<input1.Length;i++){

if(s[i]>='A' && s[i]<='Z'){

result+=(int)s[i]-64;

}

}

}

else{

for(int i=0;i<input1.Length;i++){

if((s[i]>='A' && s[i]<='Z') && (s[i]!='A') &&(s[i]!='E') && (s[i]!='I') &&(s[i]!='O') && (s[i]!='U'))

{

result+=(int)s[i]-64;

}

}

}

return result;

}

}

/\* sum of power of digits\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int sumOfPowerOfDigits(int input1)

{

//Read only region end

//Write code here

int[] intarr = new int[10];

int i = 0;

int sum = 1;

while (input1 > 0)

{

intarr[i] = input1 % 10;

input1 /= 10;

i++;

}

for (int j = i; j > 0; j--)

{

sum = sum + (int)Math.Pow(intarr[j], intarr[j - 1]);

}

return sum;

}

}

/\* simple encoded array\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

//Assume following return types while writing the code for this question.

public class Result

{

public int output1;

public int output2;

}

public Result findOriginalFirstAndSum(int[] input1, int input2)

{

//Read only region end

//Write code here

int first = input1[input2 - 1];

int sum = input1[input2 - 1];

for (int i = input2 - 2; i >= 0; i--)

{

first = input1[i] - first;

sum = sum + first;

}

Result r = new Result();

r.output1 = first;

r.output2 = sum;

return r;

}

}

/\*most freq digit\*/

using System;

using System.Collections.Generic;

//using System.Linq;

//Read only region start

public class UserMainCode

{

public int mostFrequentlyOccurringDigit(int[] input1, int input2)

{

//Read only region end

//Write code here

int[] arr = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

int max = 0, index = 0;

for (int i = 0; i < input2; i++)

{

do

{

int rem = input1[i] % 10;

arr[rem]++;

input1[i] = input1[i] / 10;

} while (input1[i] > 0);

}

for (int j = 0; j < arr.Length; j++)

{

if (arr[j] >= max)

{

max = arr[j];

index = j;

}

}

return index;

}

}

/\* long dec seq\*/

using System;

using System.Collections.Generic;

// Read only region start

class UserMainCode

{

public class Result

{

public final int output1;

public final int output2;

public Result(int out1, int out2)

{

output1 = out1;

output2 = out2;

}

}

public Result decreasingSeq(int[] input1, int input2)

{

// Read only region end

//Write code here...

int no\_of\_seq = 0;

int long\_seq = 0, c = 1;

for (int i = 1; i < input2; i++)

{

if (input1[i] < input1[i - 1])

{

c++;

}

else

{

if (c > long\_seq && c != 1)

{

long\_seq = c;

}

if (c > 1)

{

no\_of\_seq++;

}

c = 1;

}

}

if (c > long\_seq && c != 1)

{

long\_seq = c;

}

if (c > 1)

{

no\_of\_seq++;

}

Result r = new Result(no\_of\_seq, long\_seq);

return r;

}

}

/\* get code through strings\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int getCodeThroughStrings(string input1)

{

//Read only region end

//Write code here

string[] strarr1 = input1.Split(' ');

int sum = 0;

int rsum = 0;

for (int i = 0; i < strarr1.Length; i++)

{

sum = sum + strarr1[i].Length;

}

if (sum > 9)

{

int a = sum % 10;

sum = sum / 10;

rsum = sum + a;

}

return rsum;

}

}

/\* find string code\*/

using System;

using System.Collections.Generic;

//Read only region start

public class UserMainCode

{

public int findStringCode(string input1)

{

//Read only region end

//Write code here

int r = 0;

string[] str2 = new string[10];

string stri = input1.ToUpper();

string[] str = stri.Split(' ');

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

{

char[] carr = str[i].ToCharArray();

int sum = 0;

for (int j = 0; j < carr.Length / 2; j++)

{

sum = sum + Math.Abs(GetValue(carr[j]) - GetValue(carr[carr.Length - 1 - j]));

}

if ((carr.Length % 2) != 0)

{

sum = sum + GetValue(carr[carr.Length / 2]);

}

str2[r++] = sum.ToString();

}

string st = "";

for (int k = 0; k < r; k++)

{

st += str2[k];

}

return int.Parse(st);

}

public int GetValue(char c)

{

return (int)c - 64;

}

}

/\* hill range\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

//input1 = no. of rows; input2 = initial value at 1st row; input3 = increment of initial value by input3

class hillRange

{

static int hillRan(int input1,int input2,int input3)

{

int temp = input2, x = 0, hill = 0;

for (int i = 1; i < input1; i++)

{

x = input2 + input3;

for (int j = 0; j <= i; j++)

{

hill += x;

}

input2 = x;

}

return (temp + hill);

}

static void Main(string[] args)

{

Console.WriteLine(hillRan(5,10,2));

Console.ReadKey();

}

}

}

/\* nth prime\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class nPrime

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int num = 1, i;

int count = 0;

while (count < n)

{

num = num + 1;

for ( i = 2; i <= num; i++)

{

if (num % i == 0)

{

break;

}

}

if (i == num)

{

count = count + 1;

}

}

Console.WriteLine("Your nth prime is: " + num);

Console.ReadKey();

}

}

}

/\* n prime2\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class nPrime2

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int c = 0;

int i = 1;

while(n>0)

{

i = i + 1;

for (int j = 2; j <= i/2; j ++)

{

if(i%j == 0)

{

c++;

break;

}

}

if(c == 0)

{

n = n - 1;

}

c = 0;

}

Console.WriteLine(i);

Console.ReadKey();

}

}

}

/\* possible palilndromes\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class palinPossible

{

static int palin(int input1)

{

if (input1 <= 9)

{

return 2;

}

int temp = input1, c = 0;

int l = input1.ToString().Length;

int[] a = new int[l];

for (int i = 0; i < l; i++)

{

a[i] = temp % 10;

temp /= 10;

}

Array.Sort(a);

if (input1 <= 99)

{

if (a[0] == a[1])

{

return 2;

}

}

if (input1 <= 999)

{

if (a[0] == a[1] || a[0] == a[2] || a[1] == a[2])

return 2;

}

if (input1 <= 9999)

{

if ((a[0] == a[1]) && (a[2] == a[3]))

return 2;

}

if (input1 > 9999)

{

for (int i = 0; i < l; i++)

{

for (int j = i + 1; j < l; j++)

{

if (a[i] == a[j])

{

c++;

i = i + 1;

}

}

}

if (c == 2)

{

return 2;

}

}

return 1;

}

static void Main(string[] args)

{

int i = int.Parse(Console.ReadLine());

Console.WriteLine(palin(i));

}

}

}

/\*pin\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class pin

{

static int res(int input1,int input2,int input3)

{

int p = 0, i = 0, temp = 0, l = 0;

int a = input1, b = input2, c = input3;

int[] num = new int[9];

//int[] l = new int[3];

while (a > 0)

{

num[i] = a % 10;

a /= 10;

i++;

}

while (b > 0)

{

num[i] = b % 10;

b /= 10;

i++;

}

while (c > 0)

{

num[i] = c % 10;

c /= 10;

i++;

}

l = Math.Min(num[2], num[5]);

temp = (temp \* 10) + Math.Min(l, num[8]);

l = Math.Min(num[1], num[4]);

temp = (temp \* 10) + Math.Min(l, num[7]);

l = Math.Min(num[0], num[3]);

temp = (temp \* 10) + Math.Min(l, num[6]);

Array.Sort(num);

p = ((num[8] \* 1000) + temp);

return p;

}

static void Main(string[] args)

{

Console.WriteLine(res(190,267,853));

}

}

}

/\* prime range\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class primeRange

{

static void Main(string[] args)

{

int c = 0, count = 0;

int input1 = int.Parse(Console.ReadLine());

int input2 = int.Parse(Console.ReadLine());

for (int i = input1; i <= input2; i++)

{

c = 0;

for (int j = 2; j <= i / 2; j++)

{

if (i % j == 0)

{

++c;

break;

}

}

if (c == 0)

{

++count;

}

}

Console.WriteLine(count);

Console.ReadKey();

}

}

}

/\* string palindrome\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace stringDemo

{

class strPalin

{

static int palin(string input1)

{

input1 = input1.ToLower();

string res = "";

char[] s = new char[input1.Length];

int j = 0;

for (int i = input1.Length - 1; i >= 0; i--)

{

s[i] = input1[j];

j++;

}

for(int i = 0; i < input1.Length; i ++)

{

res += s[i];

}

if (input1.Equals(res))

{

return 2;

}

return 1;

}

static void Main(string[] args)

{

Console.WriteLine(palin("MADam"));

Console.ReadKey();

}

}

}

/\* ones tens 100s\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace units\_hundreds\_place

{

class Program

{

static void Main(string[] args)

{

int i = 1;

Console.WriteLine("enter value of n1");

int n1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter value of n2");

int n2 = Convert.ToInt32(Console.ReadLine());

while (n1 > 0)

{

int rem = n1 % 10;

if (rem == n2)

{

break;

}

n1 = n1 / 10;

i++;

}

switch (i)

{

case 1: Console.WriteLine("{0} is in units place",n2);

break;

case 2:

Console.WriteLine("{0} is in tens place", n2);

break;

case 3:

Console.WriteLine("{0} is in hundreds place", n2);

break;

case 4:

Console.WriteLine("{0} is in thousands place", n2);

break;

}

Console.ReadKey();

}

}

}