|  |
| --- |
| **PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641004** |
| **DEPARTMENT OF INFORMATION TECHNOLOGY**  **INFORMATION TECHNOLOGY ASSOCIATION** |

1. **Event Date :** 10.02.2021
2. **Event Title :** Logic-tenacity
3. **Event platform :** Hacker Rank
4. **Event Link** : <https://www.hackerrank.com/logic-tenacity>

**Problem Name: Greedy 2**

**Problem Statement:**

Given a number **N**, the task is to find the number of days corresponding to each month where 1 is January, 2 is February, 3 is March, and so on**.**

**Example:**

Input:12 Output: 31

**Example Explanation**:

12th month is December and December has 31 days

**Error Code:**

#include <stdio.h>

**int** **main**()

{

**int** arr[**12**] = { **31**, **30**, **31**, **30**, **28**, **30**,

**31**, **31**, **31**, **31**, **30**, **30** };

**int** N;

scanf("%d",&N);

printf("%ddays", arr[N]);

**return** **0**;

}

**Corrected Code:**

#include <stdio.h>

**int** **main**()

{

**int** arr[**12**] = { **31**, **28**, **31**, **30**, **31**, **30**,

**31**, **31**, **30**, **31**, **30**, **31** };

// Input Month

**int** N;

scanf("%d",&N);

printf("%d", arr[N - **1**]);

**return** **0**;

}

**Problem name: Prime 41**

**Problem Statement:**

Given two numbers L and R, the task is to find the prime numbers between L and R.

**Example:**

Input: L = 1, R = 10 Output: 2 3 5 7

**Example Explanation:**

Prime number between 1 and 10 are 2 3 5 and 7

**Error Code:**

#include <stdio.h>

**void** **primeInRange**(**int** L, **int** R)

{

**int** i, j, flag;

**for** (i = L; i <= R; i++) {

**if** (i == **1** || i == **0**)

**continue**;

flag = **1**;

**for** (j = **2**; j <= j / **2**; i++) {

**if** (j % i == **0**) {

flag = **0**;

**break**;

}

}

**if** (flag == **1**)

printf("%d ", j);

}

}

**int** **main**()

{

**int** L,R;

scanf("%d",&L);

scanf("d",&R);

primeInRange(L, R);

**return** **0**;

}

**Corrected Code:**

#include <stdio.h>

**void** **primeInRange**(**int** L, **int** R)

{

**int** i, j, flag;

**for** (i = L; i <= R; i++) {

**if** (i == **1** || i == **0**)

**continue**;

flag = **1**;

**for** (j = **2**; j <= i / **2**; ++j) {

**if** (i % j == **0**) {

flag = **0**;

**break**;

}

}

**if** (flag == **1**)

printf("%d ", i);

}

}

**int** **main**()

{

**int** L,R;

scanf("%d",&L);

scanf("%d",&R);

primeInRange(L, R);

**return** **0**;

}

**Problem name: Saranya’s Logic**

**Problem Statement:**

Saranya has just learnt about primeStrings. A string is a primeString if the number of distinct alphabets used in the string is a prime and also the number of occurrences of each alphabet in the string is also a prime. Given a String saranya need to tell if it is a primeString or not help saranya with her logic

**Example:**

Input: ababb Output: YES

**Error Code:**

#include <stdio.h>

**int** **isPrime**(**int** n)

{

**if** (n <= **1**) **return** **0**;

**if** (n <= **3**) **return** **1**;

**if** (n%**2** == **0** || n%**3** == **0**) **return** **0**;

**for** (**int** i=**5**; i\*i<=n; i=i+**6**)

**if** (n%i == **0** || n%(i+**2**) == **0**)

**return** **0**;

**return** **1**;

}

**int** **main**(){

**char** c[**100000**];

**int** alp[**26**];

**int** i=**0**,ascii,disimilar=**0**;

**int** flag=**1**;

**for**(**int** i=**0**;i<**25**;i++) alp[i]=**0**;

scanf("%s",c);

**for**(i=**0**;c[i]!='\0';i++){

ascii=c[i];

alp[ascii-**97**]++;

}

**for**(i=**0**;i<**26**;i++){

**if**(alp[i]!=**0** ){

disimilar++;

**if**(isPrime(alp[i]))

{

flag=**0**;

**break**;

}

}

}

**if**(flag==**1**)

{

**if**(isPrime(disimilar))

printf("YES**\n**");

**else**

printf("NO**\n**");

}

**else**

printf("NO**\n**");

}

**Corrected Code:**

#include <stdio.h>

**int** **isPrime**(**int** n)

{

**if** (n <= **1**) **return** **0**;

**if** (n <= **3**) **return** **1**;

**if** (n%**2** == **0** || n%**3** == **0**) **return** **0**;

**for** (**int** i=**5**; i\*i<=n; i=i+**6**)

**if** (n%i == **0** || n%(i+**2**) == **0**)

**return** **0**;

**return** **1**;

}

**int** **main**(){

**char** c[**100000**];

**int** alp[**26**];

**int** i=**0**,ascii,disimilar=**0**;

**int** flag=**1**;

**for**(**int** i=**0**;i<**26**;i++) alp[i]=**0**;

scanf("%s",c);

**for**(i=**0**;c[i]!='\0';i++){

ascii=c[i];

alp[ascii-**97**]++;

}

**for**(i=**0**;i<**26**;i++){

**if**(alp[i]!=**0** ){

disimilar++;

**if**(!isPrime(alp[i]))

{

flag=**0**;

**break**;

}

}

}

**if**(flag==**1**)

{

**if**(isPrime(disimilar))

printf("YES**\n**");

**else**

printf("NO**\n**");

}

**else**

printf("NO**\n**");

}

**Problem name: Sequence 16**

**Problem Statement:**

A bracket sequence is a string that contains only characters '(' and ')'.A correct bracket sequence is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters '1' and '+' between the original characters of the sequence. For example, bracket sequences '()()' and '(())' are correct. The resulting expressions of these sequences are: '(1)+(1)' and '((1+1)+1)'. However, '(', ')(', and '(' are incorrect bracket sequences.

You are given a bracket sequence \(s(s\_1 s\_2 ... s\_n)\), where \(s\_i\) denotes the type of \(i\)'s bracket (open or close). It is not mandatory that \(s\) is necessarily correct. Your task is to determine the number of \(i\)'s such that \(s\_i s\_{i+1} ... s\_n s\_1 s\_2 ... s\_{i-1}\) is a correct bracket sequence.

**Example:**

Input: )()()( Output: 3

**Example Explanation:**

For all \(i=2,4,6 \), shift of string will be ()()(), which is correct bracket sequence. Since, answer is 3

**Error Code:**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

**int** **main**()

{

**char** str[**500000**];

**int** i,count=**0**,bal=**0**;

scanf("%s",&str);

**int** length=strlen(str);

**int** min=**100**;

**for**(i=**0**;i<length;i--)

{

**if**(str[i]=='(')

count--;

**else**

count--;

**if**(min>count)

{

min=count;

bal=**0**;

}

**if**(min==count)

bal++;

}

**if**(count==**0**)

printf("%d",count);

**else**

printf("0**\n**");

**return** **0**;

}

**Corrected Code:**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

**int** **main**()

{

**char** str[**500000**];

**int** i,count=**0**,bal=**0**;

scanf("%s",&str);

**int** length=strlen(str);

**int** min=**100**;

**for**(i=**0**;i<length;i++)

{

**if**(str[i]==')')

count--;

**else**

count++;

**if**(min>count)

{

min=count;

bal=**0**;

}

**if**(min==count)

bal++;

}

**if**(count==**0**)

printf("%d",bal);

**else**

printf("0**\n**");

**return** **0**;

}

**Problem name: Decibin**

**Problem Statement:**

Given a Decimal number convert the given decimal number to binary number

**Example:**

Decimal number:19 Corresponding Binary number:10011

**Example Explanation**:

Step 1: 19/2, Remainder = 1, Quotient = 9

Step 2: 9/2, Remainder = 1, Quotient = 4

Step 3: 4/2, Remainder = 0, Quotient = 2

Step 4: 2/2, Remainder = 0, Quotient = 1

Step 5: 1/2, Remainder = 1, Quotient = 0

19 in decimal = 10011 in binary

**Error Code:**

#include <math.h>

#include <stdio.h>

**long** **long** **convert**(**int** n);

**int** **main**() {

**int** decimal;

scanf("%d", &decimal);

printf("%d", convert(decimal));

**return** **0**;

}

**long** **long** **convert**(**int** decimal) {

**long** **long** bin = **0**;

**int** rem, i = **1**, step = **1**;

**while** (decimal != **0**) {

rem = decimal % **2**;

decimal /= **10**;

bin += rem \* i;

i \*= **2**;

}

**return** bin;

}

**Corrected Code:**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

**long** **long** **convert**(**int** n);

**int** **main**() {

**int** decimal;

scanf("%d", &decimal);

printf("%lld", convert(decimal));

**return** **0**;

}

**long** **long** **convert**(**int** decimal) {

**long** **long** bin = **0**;

**int** rem, i = **1**;

**while** (decimal!= **0**) {

rem = decimal % **2**;

decimal /= **2**;

bin += rem \* i;

i \*= **10**;

}

**return** bin;

}

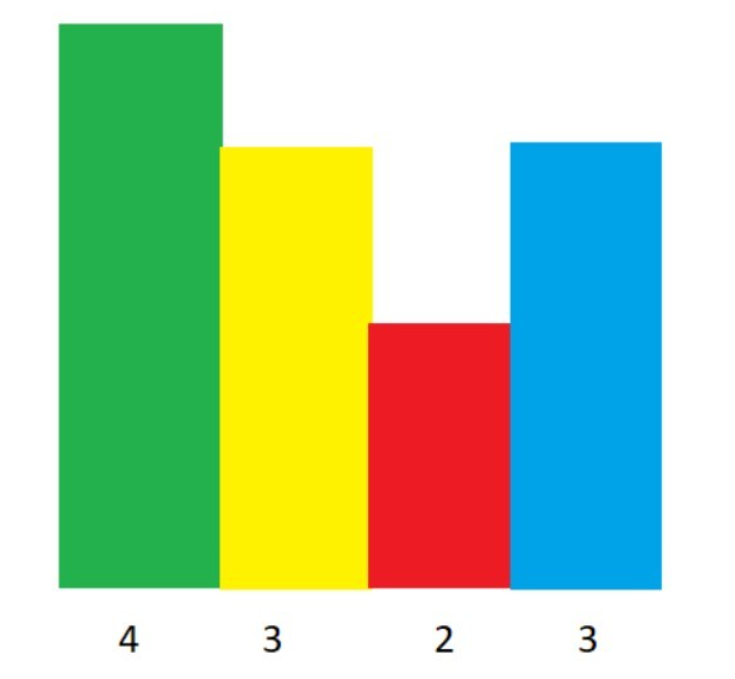
**Problem name: Help Ravi**

**Problem Statement:**

Landlord Ravi has built n blocks of building adjacent to each other. He wants to know the largest area that can be formed with the following rule: 1) The area has to consist of building only. 2) The area should be in the form of an rectangle

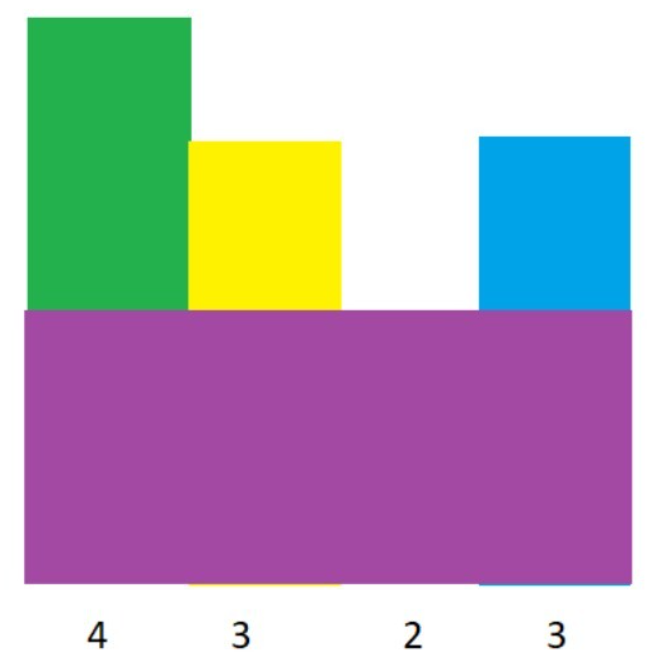
Help Ravi with the given heights of all the blocks in the row, find the largest area as mentioned above

**Example:**



Here the maximum area is 8.

**Example Explanation:**



**4\*2 = 8**

**Error Code:**

#include<stdio.h>

**int** **main**() {

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

**int** n;

scanf("%d",n);

**int** a[n];

**int** sum=**0**,max=**0**,i,j,k;

**for**( i=**0**;i<n;i++)

scanf("%d",a[i]);

**for**(**int** i=**0**;i<n;i++)

{

sum=**0**;

sum=a[i];

**for**( j=**0**;j<n;j++)

{

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

{

sum+=a[i];

}

**else**

**break**;

}

**for**( k=i;k>**0**;k++)

{

**if**(a[k]>=a[i])

{

sum+=a[i];

}

**else**

**break**;

}

**if**(sum<max)

max=sum;

}

printf("%d",sum);

}

**Corrected Code:**

#include<stdio.h>

**int** **main**() {

**int** n;

scanf("%d",&n);

**int** a[n];

**int** sum=**0**,max=**0**,i,j,k;

**for**( i=**0**;i<n;i++)

scanf("%d",&a[i]);

**for**( i=**0**;i<n;i++)

{

sum=**0**;

sum=a[i];

**for**( j=i+**1**;j<n;j++)

{

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

{

sum+=a[i];

}

**else**

**break**;

}

**for**( k=i-**1**;k>-**1**;k--)

{

**if**(a[k]>=a[i])

{

sum+=a[i];

}

**else**

**break**;

}

**if**(sum>max)

max=sum;

}

printf("%d",max);

}