**19)** Mikes likes to play with numbers. His friends are also good with numbers and often plays mathematical games. They made a small game where they will spell the last digit of a factorial of a number other than 0. Let say the given number is 5, so 5! (5 factorial) will be

5 \*4\* 3\* 2\* 1= 120.

Here 0 is the last digit. But, we don't want 0, we want a

number other than 0.

Then the last digit is 2. This is what we have to output

#include <iostream.h>

Using namespace std;

Int factorial(int n)

{

If(n<=1)

{

Return 1}

Else

{

return n\*factorial(n-1);

}

}

int main()

{

Int n;

Cin>>n;

Int fact=factorial(n);

While (fact%10==0)

{

Fact=fact/10;

}

Cout<<fact%10;

return 0;

}

**20)** An accountant in a firm has a serious issue with numbers. He types the numbers in a reverse manner. Suppose he has to enter 123, he enters the number 321. He has the habit of reading numbers from right to left.

The boss became aware of this only after he found out at the end of the billing month when he had to file the tax. He has to correct al the numbers by reentering each number from right to left. This gives the corrected number.

Given a number N, help the boss find out the corrected numbers. Display the corrected numbers as output. Also ignore any 0’s at the end of the number.

Note

The corrected numbers should be only 16-bit signed integers. If the output (the corrected number is outside the range display “Wrong value”.

#include <iostream>

#include <limits>

using namespace std;

int reverseNumber(int n) {

int reversed = 0;

while (n != 0) {

int digit = n % 10;

reversed = reversed \* 10 + digit;

n /= 10;

}

return reversed;

}

int main() {

int N;

cout << "Enter the number N: ";

cin >> N;

int corrected = reverseNumber(N);

if (corrected > numeric\_limits<short>::max() || corrected < numeric\_limits<short>::min()) {

cout << "Wrong value" << endl;

} else {

cout << "Corrected number: " << corrected << endl;

}

return 0;

}

**21)** A Company has decided to give some gifts to all of its employees. For that, the company has given some rank to each employee. Based on that rank, the company has made certain rules to distribute the gifts.

The rules for distributing the gifts are:

Each employee must receive at least one gift.

Employees having higher ranking get a greater number of gifts than their neighbours.

What is the minimum number of gifts required by the company?

**Input**

2

5

1 2 1 5 2

2

1 2

**Output**

7

3

#include <iostream>

#include <vector>

using namespace std;

int minGiftsRequired(const vector<int>& ranks) {

int n = ranks.size();

vector<int> gifts(n, 1);

// Forward pass

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

if (ranks[i] > ranks[i - 1]) {

gifts[i] = gifts[i - 1] + 1;

}

}

// Backward pass and total gifts calculation

int totalGifts = gifts[n - 1];

for (int i = n - 2; i >= 0; --i) {

if (ranks[i] > ranks[i + 1]) {

gifts[i] = max(gifts[i], gifts[i + 1] + 1);

}

totalGifts += gifts[i];

}

return totalGifts;

}

int main() {

int T;

cin >> T;

while (T--) {

int N;

cin >> N;

vector<int> ranks(N);

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

cin >> ranks[i];

}

int result = minGiftsRequired(ranks);

cout << result << endl;

}

return 0;

}

**22)** A game company has designed an online lottery game. Bingo, in this game, N number cards are displayed. Each card has a value on it. The value can be negative or positive. The player must choose two cards. To win one game, the product of the values of the two cards must be maximum value possible for any pair of cards in the display. The winning amoint will be the sum of the two cards chosen by the player.

Write an algorithm to find the winning amount as the sum of the value of the cards whose product value is maximum.

**Input**

The second line consists of N space-separated integers - val1, val2....., valN representing the values on the cards.

**Output**

Print an integer representing the sum of the values of the two cards whose product value is maximum.

**Constraints**

0 < numCards < 106

-106 < vali < 106

0 < i < numCards

**Example**

**Input:**

7

9 -3 8 -6 -7 8 10

**Output:**

19

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

long long findMaxProductSum(vector<int>& cards) {

sort(cards.begin(), cards.end());

// The maximum product can be achieved by multiplying the two largest positive numbers

long long maxProductSum = static\_cast<long long>(cards[cards.size() - 1]) + cards[cards.size() - 2];

return maxProductSum;

}

int main() {

int numCards;

cin >> numCards;

vector<int> cards(numCards);

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

cin >> cards[i];

}

long long result = findMaxProductSum(cards);

cout << result << endl;

return 0;

}  
  
**23)** A new medicine named "Medio-cine" is out in the market which helps in treating auto-immune diseases in humans. These are very common in 0-10 year old kids or greater than or equal to 81 year old senior citizens. Both these age groups are considered at a very high risk.

The government wants to distribute the medicine as soon as possible among all age groups. There is only one(1) political center from where this medicine is distributed. In a single day we cannot provide to both high risk and non high risk age groups.Each person requires only one capsule and there can be L capsules distributed in a day. The high risk humans are supposed to be considered first.There are 'N' humans and 'L' capsules per day. Find the minimum number of days required to medicate all the N humans.

**Input format**

First Line: Contains a Positive Integer denoting N and L denoting number of medicines.

Next line Contains N elements of the array A space separated denoting age.

**Constraints:**

1<=N<10000

1<=L=1000

1<=A[i]=1000

**Example 1:**

**Input:**

5 2  -> N, L

11 81 27 72 79 -> array A (age of N humans respectively)

**Output:**

3 Days

#include <iostream>

using namespace std;

int minDays(int n,int l, int\* age)

{

int days=0;

int highrisk=0;

int nonrisk=0;

int capsules;

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

if(age[i]<=11 || age[i]>=81)

highrisk++;

else

nonrisk++;

while(highrisk>0 || nonrisk>0)

{

capsules = l;

while(capsules > 0 && highrisk > 0)

{

capsules--;

highrisk--;

}

while(capsules > 0 && nonrisk > 0)

{

capsules--;

nonrisk--;

}

days++;

}

return days;

}

int main()

{

int n,l;

cin>>n>>l;

int\* age=new int[n];

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

{

cin>>age[i];

}

int res=minDays(n,l,age);

cout<<res<<" Days";

return 0;

}

**24)** Jack and Jill went to a carnival, there are many games counter in the carnival. And if you win you may get good prizes.

One of the games there was Check-and-win. The game was simple but a bit tricky. Below are the rules of the game:

• There will be a strip with N integers mentioned on it.

•Each player will get their respective number strips.Each time one player has to pick a number from the strip. The index chosen by the one player cannot be used further in the game. We will understand more in below example.

•The next player has to pick a number from another index from their respective strip. Once that index is used, it won't be used again in the game by either of the payers.

• At the end of the when all the indexes are checked. The sum will be calculated for each player.

• The player with the maximum sum will be the winner.

Jack decided a rule, that he will always start first. Print Jack if he wins, or Jill when Jill wins. If there is a tie, print Tie.

Considering that both the players are playing optimally. find the output.

Let us try to understand it with an example. Consider the strip of 3 integers. Means N = 3. Respective strip for Jack and Jill is as follows:

JK: [134]

JL: (531)

The rule says that Jack always plays first.

Step 1: Jack plays first and chooses the largest element his number-strip JK[3] = 4.

Now index 3 is checked.

Step 2: Jill plays next, now she has to find the largest number among index 1 and 2. Within these 2 indices she has the largest number at index 1. So, she chooses JL[1] Now index 1 is checked.

Step 3: Its Jack's turn now, and the only index left is 2. So, he chooses JK[2]

Sum of each:

Jack : 4 + 3 =7

Jill :5

Clearly jack wins.

Hence the answer is jack.

**Input format**

First Input -Accept value of Integer, N.

 Next 'N' Lines-Elements of sequence JK[]

Next 'N' Lines-Elements of sequence JI[]

**Output format**

The output is an integer value as per above logic. (Check the output in Example 1, Example 2).

additional messages in output will cause the failure of test cases.

**Constraints**

1<=N<=1000

1 <= Jk[], JL[]<=10000

Only integer values

**Example 1:**

**Input:**

2 ->Input integer , N

1 1->Input integer , Jk[]

2 2->Input integer , JL[]

**Output:**

Jill->output

#include<iostream>

#include<climits>

std::string findWinner(int N,int\* jackStrip, int\* jillStrip) {

bool\* visited = new bool[N+1]{false};

int jackSum = 0, jillSum = 0;

int jackIndex = 0, jillIndex = 0;

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

int maxJack = INT\_MIN,maxJill = INT\_MIN;

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

if(!visited[j+1] && jackStrip[j]>maxJack) {

maxJack = jackStrip[j];

jackIndex = j+1;

}

if(!visited[j+1] && jillStrip[j]>maxJill) {

maxJill = jillStrip[j];

jillIndex = j+1;

}

}

visited[jackIndex] = true;

visited[jillIndex] = true;

jackSum += maxJack;

jillSum += maxJill;

}

delete[] visited;

if(jackSum > jillSum) {

return "Jack";

} else if(jillSum > jackSum) {

return "Jill";

} else {

return "Tie";

}

}

int main() {

int N;

std::cin >> N;

int\* jackStrip = new int[N];

int\* jillStrip = new int[N];

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

std::cin >> jackStrip[i];

}

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

std::cin >> jillStrip[i];

}

std::string Winner = findWinner(N,jackStrip,jillStrip);

std::cout << Winner << std::endl;

delete[] jackStrip;

delete[] jillStrip;

return 0;

}

**25)** You are given a non-negative integer num`. You will also be given an array `Digits of size exactly 10 where each element of the array will be between 0-9 inclusive.

You can perform the following operation on the given integer ‘num’ at most once.

You can select a continuous part of ‘num’ and name it ‘x’. For each digit ‘D’ in the number ‘x’, you can change it to digits [D] (consider 0-based indexing).

You need to convert the given integer ‘num’ to a maximum possible number after performing the given operation at most once.

**Example 1:**

**Input:**

132 -> ‘num’

 9 8 1 2 7 6 5 4 0 3 -> Elements of Array`Digits (Size of Array 'Digits is always 10 and elements of the array are space separated)

**Output:**

832

**Explanation:**

You can change the first digit from 1 to 8, then the final number will be 832.

Digit 3 can be mapped to 2, but it will change 832 to 822 which is lesser.

So, 832 is the maximum number possible.

#include <iostream>

using namespace std;

int main() {

std::string num;

std::getline(std::cin, num);

int digits[10];

for(int i = 0;i < 10;i++) {

std::cin >> digits[i];

}

int length = num.length();

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

int digit = num[i] - '0';

if(digit < digits[digit]) {

num[i] = static\_cast<char>(digits[digit] + '0');

} else if(digit > digits[digit]) {

break;

}

}

std::cout << num << std::endl;

return 0;

}

**10)** A shopkeeper in a nearby town always starts a business ₹0. He never uses the previous days money for transaction. Any item in his shop costs ₹30.There are N number of customers waiting in the queue to buy items. A customer can buy any number of items but worth only ₹30.The customer can transact with shopkeeper only with the denominations ₹30,₹60,₹120.The task here is to find the transaction between the shopkeeper and customer is possible.

The customer should be able to buy the item.

The amount each customer uses for his transaction is given as array elements .

The shopkeeper should be able to return the exact change.

Display ‘Transaction Successful’ on the successful transaction with all the customers in the queue.

Display ‘Transaction failed’ on the unsuccessful transaction with any one customer in the queue.

****Example 1****

Sample Input:

3 -> value of N

30

30

60->a[] ,Elements a[0] to a[N-1],where input of each element is seperated by a new line.

****Sample Output:****

Transaction successful.

**11)** Given a string in which the same character occurs in many consecutive character elements. Your task is to find the characters that have even frequency and are consecutive. Display the sum of every frequency count( For even frequency only)

****Example 1:****

****Sample Input:****

aaabbaccccdd

****Sample Output:****

8

**13)** I guess you all remember that number game where you used to ask your friend to choose a number, then multiply it by some number, then add by some number, and after doing multiple operations, then the answer.

And you were successfully able to find that initial number that was chosen by your friend the beginning of the game.

We are going to make something similar to this. So, you asked your friend to choose a number N, and multiply it by a given number A.

Next, your friend has to choose a divisor of N, let's say 'Z', and add it to the above product. Now the final result becomes a number say X.

 If we frame it as an equation, it can be represented as X = AN

So, now you know the value of A and X. Your task is to find the number N, which was chosen by your friend. The values of N can be multiple, You have to print all the possible values separated by SPACE. If there is no valid value of N, then reply None. This means your friend didn't give you the correct reply.

Let us try to understand it with an example.

 Consider you have given a value of A = 3 and finally received the output value x = 35 Putting the values in equation X=A\*N+Z,we get:

 35-3\*N+Z, and Z is one of the divisors of N. With a smaller number like 1 or 2, this is not possible. So, we will start with some bigger numbers such as N = 8 Then Z can be 2 or 4 or 8, but in that case, the result will be even, but the answer is odd. So, we move to the next value 9.

With N = 9 ,Z can be 3putting this in the equation: 35 =3\*N+Z, and Z is one of the divisors of N.

3\*9+3 = 30 which doesn't matches with 35. So, we move

to the next integer.

With N = 10 , Z can be 2 or 5, putting this in the equation:

35=3\*N+Z

With Z = 2

3 + 10 + 2 = 32 still not equal to 35.

With z = 5

3\*10 + 5 = 35 , still now equal to 35.

So, one of the values of N is 10.

Likewise, if we proceed, we cannot find any other value which could satisfy the above conditions.

So, the output is only 10.

****Example 1:****

****Input:****

50 5 -> Input integer, A, X

****Output:****

****None****  ->Output

****Input format:****

First Input Accept value of Integer A.

Second Input-Accept value of Integer, K (Next Line).

****Output format :****

The output are either None or integers values (separated by space) value as per above logic. (Check the output in Example 1, Example 2).

****Constraints:****

1 <= xx = 1000000

1 <= A <= X

Only positive integer values