GROUP 1

# 1.

An array A with N (N<100) positive integers and an array B with M (M<100) positive integers are read from SI. Count all the pairs of elements that satisfy the following condition // where x is any element from the array A and y is any element from the array B. Notice: to calculate the power of a given number you can use the function pow from the library math.h.

Input”

N=3, A[]={2,1,5}

M=2, B[]={1,5}

Output: 3

Explanation: The pairs are {2,1}, {2,5} and {5,1}.

Solution

#include **<stdio.h>**#include **<math.h>  
  
int** main() {  
 **int** N;  
 scanf(**"%d"**,&N);  
 **int** A[100];  
 **for**(**int** i=0 ; i<N ; i++){  
 scanf(**"%d"**,&A[i]);  
 }  
 **int** M;  
 scanf(**"%d"**,&M);  
 **int** B[100];  
 **for**(**int** i=0 ; i<M ; i++){  
 scanf(**"%d"**,&B[i]);  
 }  
 **int** counter=0;  
 **for**(**int** i=0 ; i<N ; i++){  
 **for**(**int** j=0 ; j<M ; j++){  
 **if**(pow(A[i],B[j])> pow(B[j],A[i])){  
 counter++;  
 }  
 }  
 }  
 printf(**"%d"**,counter);  
 **return** 0;  
}

# 2.

The dimensions of a matrix (m<100 and n<100) and its integer elements are read from SI. Print all the elements from the matrix that satisfy the condition: the sum of the remaining elements int the corresponding row and column where the element is located is an odd number. The numbers should be printed from the left to right and from top to bottom.

Input

3 3

1 2 3

4 5 6

7 8 9

Output:

2 4 6 8

Explanation: 2 is in the first row, second column. The rest of the numbers tha lay in the first row are 1 and 3, and the numbers in the second column are 5 and 8, so 1+3+5+8 is odd, and 2 is printed, 4 -> 1+7+5+6 is odd…

Solution

#include **<stdio.h>  
  
int** main() {  
 **int** m,n;  
 scanf(**"%d%d"**,&m,&n);  
 **int** matrix[100][100];  
 **for**(**int** i=0 ; i<m ; i++){  
 **for**(**int** j=0 ; j<n ; j++){  
 scanf(**"%d"**,&matrix[i][j]);  
 }  
 }  
  
 **for**(**int** i=0 ; i<m ; i++){  
 **for**(**int** j=0 ; j<n ; j++){  
 **int** sum=0;  
 **for**(**int** k=0 ; k<n ; k++){  
 **if**(k!=j){  
 sum+=matrix[i][k];  
 }  
 }  
 **for**(**int** k=0 ; k<m ; k++){  
 **if**(k!=i){  
 sum+=matrix[k][j];  
 }  
 }  
 **if**(sum%2!=0){  
 printf(**"%d "**,matrix[i][j]);  
 }  
 }  
 }  
 **return** 0;  
}

# 3.

The file **sodrzina.txt** contains text written in multiple lines. Each line has no more than 80 characters and the characters are letters, special characters and digits. The first line has a row number of 1, the second has a row number of 2, etc. Write a program that will print the row number of the line with the longest subarray that contains onl digits. If the file does not contain any lines with numbers, the program should print a message **There are no such lines**. If there is more than one line with the same max number of digits, than the row of the last read is printed.

**For example:**

| **Input** | **Result** |
| --- | --- |
| never!?!odd,27,or12even365  A\_man\_a+5Plan510123a\_canal,Panama  Rise?!12te45vote5\*siR | 2 |

Solution

#include **<stdio.h>**#include **<string.h>**#include **<ctype.h>  
  
void** writeToFile() {  
 FILE \*f = fopen(**"sodrzina.txt"**, **"w"**);  
 **char** c;  
 **while**((c = getchar()) != **EOF**) {  
 fputc(c, f);  
 }  
 fclose(f);  
}  
  
**int** countOfDigits(**char** \*str){  
 **int** max, flag=1;  
 **int** i,j;  
 **for**(i=0 ; i< strlen(str) ; i++){  
 **if**(isdigit(str[i])){  
 **int** counter=0;  
 **for**(j=i*/\*or i+1 \*/* ; j< strlen(str) ; j++){  
 **if**(isdigit(str[j])){  
 counter++;  
 } **else**{  
 **break**;  
 }  
 }  
 **if**(flag){  
 max=counter;  
 flag=0;  
 } **else if**(counter>max){  
 max=counter;  
 }  
 i=j;  
 }  
 }  
 **return** max;  
}  
  
**int** main() {  
 writeToFile();  
 FILE \*f= fopen(**"sodrzina.txt"**,**"r"**);  
 **char** str[100];  
 **int** i=1;  
 **int** flag=1;  
 **int** maxRow,maxLength;  
 **while**(fgets(str,100,f)!=**NULL**){  
 **int** d= countOfDigits(str);  
 **if**(d>0){  
 **if**(flag){  
 maxLength=d;  
 maxRow=i;  
 flag=0;  
 } **else if**(d>=maxLength){  
 maxLength=d;  
 maxRow=i;  
 }  
 }  
 i++;  
 }  
 **if**(flag){  
 printf(**"No such lines"**);  
 } **else**{  
 printf(**"%d"**,maxRow);  
 }  
 fclose(f);  
 **return** 0;  
}

GROUP 2

# 1.

An unknown number of positive numbers is read from SI. The reading stops when something different than an integer is entered. Write a program that will print the number that contains the maximum count of unique digits. If there is more one such number, print the last one.

Example:

Input: 34 788 4 1234 11 67789 .

Explanation:

34 -> two unique digits

788 -> two unique digits

4 -> one unique digit

1234 -> four unique digits

11 -> one unique digit

67789 -> four unique digits

Output : 67789 as it has the most unique digits (the same as 1234 does but it appears later).

Solution

#include **<stdio.h>  
  
int** uniqueDigits(**int** n){  
 **int** tmpA[10]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};  
 **int** tmp=n;  
 **while**(tmp){  
 **int** ld=tmp%10;  
 tmpA[ld]++;  
 tmp/=10;  
 }  
 **int** counter=0;  
 **for**(**int** i=0 ; i<10 ; i++){  
 **if**(tmpA[i]!=-1){  
 counter++;  
 }  
 }  
 **return** counter;  
}  
  
**int** main(){  
 **int** n;  
 **int** flag=1,max,maxNumber;  
 **while**(scanf(**"%d"**,&n)){  
 **int** unique= uniqueDigits(n);  
 **if**(flag){  
 max=unique;  
 maxNumber=n;  
 flag=0;  
 } **else if**(unique>=max){  
 max=unique;  
 maxNumber=n;  
 }  
 }  
 printf(**"%d"**,maxNumber);  
 **return** 0;  
}

# 2.

The dimensions of a matrix (m<100 and n<100) and its integer elements are read from SI. Print all the elements from the matrix that satisfy the condition: there are at least 2 even numbers among the remaining elements int the same row and column where the element is located. The elements should be printed from the left to the right and from top to bottom.

Input

3 3

1 2 3

4 5 6

7 8 9

Output: 1 3 5 7 9

Explanation of the output:

1 (has exactly two even numbers int the same row and column – 2 and 4)

3 (has exactly two even numbers in the same row and column – 2 and 6)

5 (has exactly four even numbers in the same row and column – 2, 8, 4 and 6)

7 (has exactly two even numbers in the same row and column – 4 and 8)

9 (has exactly two even numbers in the same row and column – 6 and 8)

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 3  1 5 7  3 9 11  2 6 8 | 2 6 8 |

Solution

#include **<stdio.h>  
  
int** main() {  
 **int** m,n;  
 scanf(**"%d%d"**,&m,&n);  
 **int** matrix[100][100];  
 **for**(**int** i=0 ; i<m ; i++){  
 **for**(**int** j=0 ; j<n ; j++){  
 scanf(**"%d"**,&matrix[i][j]);  
 }  
 }  
  
 **for**(**int** i=0 ; i<m ; i++){  
 **for**(**int** j=0 ; j<n ; j++){  
 **int** counter=0;  
 **for**(**int** k=0 ; k<n ; k++){  
 **if**(k!=j){  
 **if**(matrix[i][k]%2==0){  
 counter++;  
 }  
 }  
 }  
 **for**(**int** k=0 ; k<m ; k++){  
 **if**(k!=i){  
 **if**(matrix[k][j]%2==0){  
 counter++;  
 }  
 }  
 }  
 **if**(counter>=2){  
 printf(**"%d "**,matrix[i][j]);  
 }  
 }  
 }  
  
 */\*for(int i=0 ; i<m ; i++){ This is a faster way  
 for(int j=0 ; j<n ; j++){  
 int counter=0;  
 for(int k=0 ; k<n ; k++){  
 if(counter==2){  
 break;  
 }  
 if(k!=j){  
 if(matrix[i][k]%2==0){  
 counter++;  
 }  
 }  
 }  
 for(int k=0 ; k<m ; k++){  
 if(counter==2){  
 break;  
 }  
 if(k!=i){  
 if(matrix[k][j]%2==0){  
 counter++;  
 }  
 }  
 }  
 if(counter==2){  
 printf("%d ",matrix[i][j]);  
 }  
 }  
 }\*/* **return** 0;  
}

# 3.

The file **sodrzina.txt** contains text written in multiple lines. Each line has no more than 80 characters and the characters are letters, special characters and digits. The first line has a row number of 1, the second has a row number of 2, etc. Write a program that will print the row number of the line with the longest subarray that contains onl letters. If the file does not contain any lines with letters, the program should print a message **There are no such lines**. If there is more than one line with the same max number of letters, than the row number of the last read is printed.

**For example:**

| **Input** | **Result** |
| --- | --- |
| never!?!odd,27,or12even365  A\_man\_a+5Plan510123a\_canal,Panama  Rise?!12te45vote5\*siR | 2 |

# **Solution**

#include **<stdio.h>**#include **<string.h>**#include **<ctype.h>  
  
void** writeToFile() {  
 FILE \*f = fopen(**"sodrzina.txt"**, **"w"**);  
 **char** c;  
 **while**((c = getchar()) != **EOF**) {  
 fputc(c, f);  
 }  
 fclose(f);  
}  
  
**int** countOfLetters(**char** \*str){  
 **int** max, flag=1;  
 **int** i,j;  
 **for**(i=0 ; i< strlen(str) ; i++){  
 **if**(isalpha(str[i])){  
 **int** counter=0;  
 **for**(j=i*/\*or i+1 \*/* ; j< strlen(str) ; j++){  
 **if**(isalpha(str[j])){  
 counter++;  
 } **else**{  
 **break**;  
 }  
 }  
 **if**(flag){  
 max=counter;  
 flag=0;  
 } **else if**(counter>max){  
 max=counter;  
 }  
 i=j;  
 }  
 }  
 **return** max;  
}  
  
**int** main() {  
 writeToFile();  
 FILE \*f= fopen(**"sodrzina.txt"**,**"r"**);  
 **char** str[100];  
 **int** i=1;  
 **int** flag=1;  
 **int** maxRow,maxLength;  
 **while**(fgets(str,100,f)!=**NULL**){  
 **int** d= countOfLetters(str);  
 **if**(d>0){  
 **if**(flag){  
 maxLength=d;  
 maxRow=i;  
 flag=0;  
 } **else if**(d>=maxLength){  
 maxLength=d;  
 maxRow=i;  
 }  
 }  
 i++;  
 }  
 **if**(flag){  
 printf(**"No such lines"**);  
 } **else**{  
 printf(**"%d"**,maxRow);  
 }  
 fclose(f);  
 **return** 0;  
}