PROBLEMS

# 1.

Write a program that for two arrays read from SI will check if they are equal. Print out the result from the comparison. The maximum size of arrays is 100.

**2.**

Write a program that for an array read from SI, will compute the sum of even elements, the sum of odd elements and will compute the ratio even/odd.

**3.**

Write a program that will compute the scalar product of two vectors with n coordinates. The number of coordinates n and the coordinates are read from SI. Print the result on the SO.

**4.**

Write a program that will check if a given array with n elements read from SI is ascending, descending or neither. Print the result.

**5.**

Write a program that will rotate elements of an array for one place in right.

**6.**

Write a program that will rotate elements of an array for m places to the right.

**7.**

Write a program that will remove duplicate from an array. After the transformation print the array

**8.**

A number N and N elements of an array are read from SI.

The array should be transformed in a new array where all the even elements should be increased by 1, and all the odd elements decreased by 1.

The transformed array should be printed to standard output and all the elements of the array should be separated with space between them.

**9.**

An array of integers A with length N (N<=100) is read from SI. Also, a integer K is read from SI.

Your task is to transform the array into a new array where the elements less than K will be first, and after them the elements greater or equal than K should follow. The order of the elements should not be changed.

Print the elements of the transformed array on standard output. All elements should be separated with a space between them.

Example:

A[] = {1, 3, 2, 5, 9, 0, 8, 10}, K=6

The transformed array will be:

A[] = {1, 3, 2, 5, 0, 9, 8, 10}

A[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0}, K=6

The transformed array will be:

A[] = {5, 4, 3, 2, 1, 0, 10, 9, 8, 7, 6}

A[]={2, 4, 6, 8, 10}, K = 5

The transformed array will be:

A[] = {2, 4, 6, 8, 10}

A[]={1, 3, 5, 7, 9} K = 5

The transformed array will be:

A[] = {1, 3, 5, 7, 9}

**10.**

An integer N is read from SI. Afterwards N arrays of integers (with maximum length of 100 elements) are read from SI. For each array, firstly the size of the array is read, and then the elements of the array.

Your task is to write a program which for each array read from SI will print the percentage of **equal symmetric numbers** in the array. Symmetric elements in an array are the first and the last, the second and element before the last, etc.

If the array has odd number of elements, then the middle element is considered both symmetric and equal.

Input:

3

5 - 7 2 5 2 8

6 - 1 2 3 3 1 1

8 - 8 4 8 3 2 1 4 8

Output and explanation:

60.00% (2,2,5 are symmetrical and equal -> 3/5 elements -> 60%)

66.67% (1,1,3,3 are symmetrical and equal -> 4/6 elements -> 66.67%)

50.00% (8,8,4,4 are symmetrical and equal -> 4/8 elements -> 50.00%)

**11.**

SOLUTIONS

# 1.

*/\* Write a program that for two arrays read from SI will check if they are equal.  
 Print out the result from the comparison. The maximum size of arrays is 100.\*/*#include **<stdio.h>  
  
int** main() {  
 **int** n,m;  
 scanf(**"%d"**,&n);  
 **int** array1[100], array2[100];  
  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array1[i]);  
 }  
  
 scanf(**"%d"**,&m);  
  
 **for**(**int** i=0 ; i<m ; i++){  
 scanf(**"%d"**,&array2[i]);  
 }  
 **if**(n!=m){  
 printf(**"Not equal"**);  
 **return** 0;  
 }  
 **for**(**int** i=0 ; i<n ; i++){  
 **if**(array1[i]!=array2[i]){  
 printf(**"Not equal"**);  
 **return** 0;  
 }  
 }  
 printf(**"Equal"**);  
 **return** 0;  
}

# 2.

*/\* Write a program that for an array read from SI, will compute the sum of even elements,  
 the sum of odd elements and will compute the ratio even/odd.\*/*#include **<stdio.h>  
  
int** main() {  
 **int** n;  
 scanf(**"%d"**,&n);  
  
 **int** sumEven=0, sumOdd=0, countEven=0, countOdd=0;  
 **int** array[100];  
  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
 **for**(**int** i=0 ; i<n ; i++){  
 **if**(array[i]%2==0){  
 sumEven+=array[i];  
 countEven++;  
 }  
 **if**(array[i]%2!=0){  
 sumOdd+=array[i];  
 countOdd++;  
 }  
 }  
  
 printf(**"Sum of even elements = %d\nSum of odd elements = %d\nEven/Odd = %.2f"**,sumEven,sumOdd,(**float**)countEven/countOdd);  
 **return** 0;  
}

# 3.

*/\*Write a program that will compute the scalar product of two vectors with n coordinates.  
The number of coordinates n and the coordinates are read from SI.  
Print the result on the SO.Write a program that will compute the scalar product of two vectors with n coordinates.  
The number of coordinates n and the coordinates are read from SI. Print the result on the SO.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** n;  
 scanf(**"%d"**,&n);  
 **int** vector1[100],vector2[100];  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&vector1[i]);  
 }  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&vector2[i]);  
 }  
 **int** scalaProduct=0;  
 **for**(**int** i=0 ; i<n ; i++){  
 scalaProduct+=vector1[i]\*vector2[i];  
 }  
 printf(**"%d"**,scalaProduct);  
 **return** 0;  
}

# 4.

*/\*Write a program that will check if a given array with n elements read from SI is ascending, descending or neither. Print the result.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** n;  
 scanf(**"%d"**,&n);  
 **int** array[100];  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
 **int** ascending=0,descending=0;  
 **for**(**int** i=0 ; i<n-1 ; i++){  
 **if**(array[i]<array[i+1]){  
 ascending++;  
 } **else if**(array[i]>array[i+1]){  
 descending++;  
 }  
 }  
 **if**(ascending==n-1){  
 printf(**"Ascending"**);  
 } **else**{  
 **if**(descending==n-1){  
 printf(**"Descending"**);  
 } **else**{  
 printf(**"Neither"**);  
 }  
 }

**return** 0;

}

# 5.

*/\*Write a program that will rotate elements of an array for one place in right.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** n;  
 scanf(**"%d"**,&n);  
 **int** array[100];  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
 **int** tmp=array[n-1];  
 **for**(**int** i=n-1 ; i>0 ; i--){  
 array[i]=array[i-1];  
 }  
 array[0]=tmp;  
 **for**(**int** i=0 ; i<n ; i++){  
 printf(**"%d "**,array[i]);  
 }  
 **return** 0;  
}

# 6.

*/\*Write a program that will rotate elements of an array for one place in right.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** n,m;  
 scanf(**"%d"**,&n);  
 **int** array[100];  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
 scanf(**"%d"**,&m);  
 **int** tmpArray[100];  
  
 *//for(int i=n-m ; i<n ; i++){  
 // tempArray[i-n+m]=array[i];  
 // }* **for**(**int** i=n-m, j=0 ; i<n ; i++, j++){  
 tmpArray[j]=array[i];  
 }  
 **for**(**int** i=n-1 ; i>=m ; i--){  
 array[i]=array[i-m];  
 }  
 **for**(**int** i=0 ; i<m ; i++){  
 array[i]= tmpArray[i];  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 printf(**"%d "**,array[i]);  
 }  
 **return** 0;  
}

Another method

*/\*Write a program that will rotate elements of an array for m places to the right.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** array[100], tempArray[100];  
 **int** n,m;  
 scanf(**"%d"**,&n);  
  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
  
 scanf(**"%d"**,&m);  
  
 **int** NewArray[100];  
  
 **for**(**int** i=0 ; i<n ; i++){  
 NewArray[(i+m)%n]=array[i];  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 printf(**"%d "**,NewArray[i]);  
 }  
 **return** 0;  
}

# 7.

*/\*Write a program that will rotate elements of an array for one place in right.\*/*#include **<stdio.h>  
  
int** main(){  
 **int** array[100];  
 **int** n;  
 scanf(**"%d"**,&n);  
  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 **for**(**int** j=i+1 ; j<n ; j++){  
 **if**(array[i] == array[j]){  
 **for**(**int** k=j ; k<n-1 ; k++){  
 array[k]=array[k + 1];  
 }  
 --n;  
 }  
 }  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 printf(**"%d "**, array[i]);  
 }  
  
 **return** 0;  
}

# 8.

*/\*A number N and N elements of an array are read from SI.  
The array should be transformed in a new array where all the even elements should be increased by 1, and all the odd elements decreased by 1.  
The transformed array should be printed to standard output and all the elements of the array should be separated with space between them.  
\*/*#include **<stdio.h>  
  
int** main(){  
 **int** array[100];  
 **int** n;  
 scanf(**"%d"**,&n);  
  
 **for**(**int** i=0 ; i<n ; i++){  
 scanf(**"%d"**,&array[i]);  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 **if**(array[i]%2==0){  
 array[i]++;  
 }**else**{  
 array[i]--;  
 }  
 }  
  
 **for**(**int** i=0 ; i<n ; i++){  
 printf(**"%d "**, array[i]);  
 }  
  
 **return** 0;  
}

# 9.

*/\*An array of integers A with length N (N<=100) is read from SI. Also, a integer K is read from SI.  
Your task is to transform the array into a new array where the elements less than K will be first,  
and after them the elements greater or equal than K should follow. The order of the elements should not be changed.  
Print the elements of the transformed array on standard output. All elements should be separated with a space between them.  
Example:  
A[] = {1, 3, 2, 5, 9, 0, 8, 10}, K=6  
The transformed array will be:  
A[] = {1, 3, 2, 5, 0, 9, 8, 10}  
  
A[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0}, K=6  
The transformed array will be:  
A[] = {5, 4, 3, 2, 1, 0, 10, 9, 8, 7, 6}  
  
A[]={2, 4, 6, 8, 10}, K = 5  
The transformed array will be:  
A[] = {2, 4, 6, 8, 10}  
  
A[]={1, 3, 5, 7, 9} K = 5  
The transformed array will be:  
A[] = {1, 3, 5, 7, 9}  
\*/*#include **<stdio.h>  
  
int** main(){  
 **int** A[100];  
 **int** N;  
 scanf(**"%d"**,&N);  
  
 **for**(**int** i=0 ; i < N ; i++){  
 scanf(**"%d"**,&A[i]);  
 }  
 **int** K;  
 scanf(**"%d"**,&K);  
 **int** smallerK[100],greaterK[100];  
 **int** countS=0,countG=0;  
 **for**(**int** i=0 ; i<N ; i++){  
 **if**(A[i]<K){  
 smallerK[countS]=A[i];  
 countS++;  
 } **else**{  
 greaterK[countG]=A[i];  
 countG++;  
 }  
 }  
 **for**(**int** i=0 ; i<countS ; i++){  
 printf(**"%d "**,smallerK[i]);  
 }  
 **for**(**int** i=0 ; i<countG ; i++){  
 printf(**"%d "**,greaterK[i]);  
 }  
 **return** 0;  
}

# 10.

*/\*An integer N is read from SI. Afterwards N arrays of integers (with maximum length of 100 elements)  
are read from SI. For each array, firstly the size of the array is read, and then the elements of the array.  
Your task is to write a program which for each array read from SI will print the percentage of equal symmetric numbers in the array.  
Symmetric elements in an array are the first and the last, the second and element before the last, etc.  
If the array has odd number of elements, then the middle element is considered both symmetric and equal.  
Input:  
3  
5 - 7 2 5 2 8  
6 - 1 2 3 3 1 1  
8 - 8 4 8 3 2 1 4 8  
Output and explanation:  
60.00% (2,2,5 are symmetrical and equal -> 3/5 elements -> 60%)  
66.67% (1,1,3,3 are symmetrical and equal -> 4/6 elements -> 66.67%)  
50.00% (8,8,4,4 are symmetrical and equal -> 4/8 elements -> 50.00%)  
\*/*#include **<stdio.h>  
  
int** main(){  
 **int** N;  
 scanf(**"%d"**,&N);  
  
 **for**(**int** i=0 ; i<N; i++){  
 **int** size;  
 scanf(**"%d"**,&size);  
 **int** A[100];  
 **for**(**int** j=0 ; j<size ; j++){  
 scanf(**"%d"**,&A[j]);  
 }  
 **int** countSym=0;  
 **for**(**int** j=0,k=size-1 ; j<size ; j++,k--){  
 **if**(A[j]==A[k]){  
 countSym++;  
 }  
 }  
 **float** percentage=(**float**)(countSym\*100)/(**float**)size;  
 printf(**"%.2f%%\n"**,percentage);  
 }  
 **return** 0;  
}