**ASSIGNMENT 4: ARRAY**

1. Write a program in C to reverse the contents of the elements of an integer array

Code:

#include<stdio.h>

int main(){

int n;

printf("Enter the size of the array");

scanf("%d",&n);

int a[n];

printf("Enter the array:\n");

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

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

int temp=0;

for (int i=0;i<(n+1)/2;i++){

temp=a[i]+a[n-1-i];

a[n-1-i]=temp-a[n-1-i];

a[i]=temp-a[i];

temp=0;

}

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

printf("%d",a[j]);

printf("\n");

return 0;

}

Output:

Enter the size of the array4

Enter the array:

1 3 5 4

4531

1. Write a program in C to read n number of values in an array. After that, count the total number of duplicate elements in that array. Then copy the elements except the duplicate elements of that array into another array and display this array in reverse order.

Code:

#include<stdio.h>

int main(){

int n;

printf("Enter the size of the array : ");

scanf("%d",&n);

int a[n];

printf("Enter the array:\n");

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

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

//Finding total number of duplicate elements

int b[1000]={0},dup=0,c[100]={0},k=0;

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

b[a[j]]+=1;

if (b[a[j]]>1){

dup+=1;

}

else{

c[k]=a[j];

k++;

}

}

printf("Total number of duplicate elements : %d\n",dup);

//Printing the unique array in reverse order

int temp=0;

for (int i=0;i<(k+1)/2;i++){

temp=c[i]+c[k-1-i];

c[k-1-i]=temp-c[k-1-i];

c[i]=temp-c[i];

temp=0;

}

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

printf("%d,",c[j]);

printf("\n");

return 0;

}

Output:

Enter the size of the array : 5

Enter the array:

1 2 1 3 2

Total number of duplicate elements : 2

3,2,1,

1. Write a menu-driven program for accepting values in two square matrices of 3x3 dimension and generate their sum, difference and product.

Code:

#include<stdio.h>

int main(){

int a[3][3]={{0}};

printf("Enter the first matrix\n");

for (int i=0;i<3;i++){

for (int j=0;j<3;j++)

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

}

printf("Enter the second matrix\n");

int b[3][3]={{0}};

for (int i=0;i<3;i++){

for (int j=0;j<3;j++)

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

}

printf("Do you want to sum(s), difference(d), product(p) ?");

int c[3][3]={{0}};

int ip;

getchar();

while ((ip=getchar())!= EOF && ip !='\n'){

switch(ip){

case 's':

for (int i=0;i<3;i++){

for (int j=0;j<3;j++){

c[i][j]+=(a[i][j]+b[i][j]);

}

}

break;

case 'd':

for (int i=0;i<3;i++){

for (int j=0;j<3;j++){

c[i][j]+=(a[i][j]-b[i][j]);

}

}

break;

case 'p':

for (int i=0;i<3;i++){

for (int j=0;j<3;j++){

for (int k=0;k<3;k++)

c[i][j]+=(a[i][k]\*b[k][j]);

}

}

break;

default : printf("Invalid choice");return 0;break;

}

}

for (int i=0;i<3;i++){

for (int j=0;j<3;j++){

printf("%d ",c[i][j]);

}

printf("\n");

}

return 0;

}

Output:

Enter the first matrix

1 2 3

4 5 6

7 8 9

Enter the second matrix

1 0 0

0 1 0

0 0 1

Do you want to sum(s), difference(d), product(p) ?s

2 2 3

4 6 6

7 8 10

Do you want to sum(s), difference(d), product(p) ?d

0 2 3

4 4 6

7 8 8

Do you want to sum(s), difference(d), product(p) ?p

1 2 3

4 5 6

7 8 9

1. Write a program to find the range of a set of integers entered by the user. Range is the difference between the smallest and biggest number in the list.

Code:

#include<stdio.h>

int main(){

int n,a[100]={0};

printf("Enter the size of the array : ");

scanf("%d",&n);

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

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

}

// Finding max element

int max=a[0];

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

if (a[i]>max)

max=a[i];

}

//Finding the minimum element

int min=a[0];

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

if (a[i]<min)

min=a[i];

}

//Finding the range

int range=0;

range=max-min;

printf("Range is : %d\n",range);

return 0;

}

Output:

Enter the size of the array : 5

1 3 5 6 8

Range is : 7

1. Write a C program which accepts ten integers from the user and prints them in ascending order. Use an array to store the integers.

Code:

#include<stdio.h>

int main(){

int n,a[100]={0};

printf("Enter the size of the array : ");

scanf("%d",&n);

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

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

}

//Basic Selection Sort Application

int min,min\_index,temp;

for (int i=0;i<n-1;i++){

min=a[i];

min\_index=i;

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

if (a[j]<min){

min=a[j];

min\_index=j;

}

}

temp=a[i]+a[min\_index];

a[min\_index]=temp-a[min\_index];

a[i]=temp-a[i];

}

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

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

}

printf("\n");

return 0;

}

Output:

Enter the size of the array : 10

1 8 9 4 2 3 6 7 5 0

0 1 2 3 4 5 6 7 8 9

1. Write a C program which accepts roll numbers of ten students and marks obtained by them in five subjects and prints the names of the students who have obtained highest and second highest marks subject wise.

Code:

#include<stdio.h>

int main(){

int a[7][10]={{0}};

printf("Enter the roll numbers: ");

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

scanf("%d",&a[0][i]);

}

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

printf("Enter the marks in 5 subjects for %d: ",a[0][i]);

for (int j=1;j<6;j++){

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

}

}

int max,submax,top\_index,subtop\_index;

for (int i=1;i<6;i++){

submax=max=a[i][0];

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

if (a[i][j]>max){

max=a[i][j];

top\_index=j;

}

}

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

if (max>a[i][j] && a[i][j]>submax){

submax=a[i][j];

subtop\_index=j;

}

}

printf("Highest marks in subject %d is received by ",i);

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

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

printf("%d ",a[0][j]);

}

}

printf("\n");

printf("Second highest marks in subject %d is received by ",i);

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

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

printf("%d ",a[0][j]);

}

}

printf("\n");

}

return 0;

}

Output:

Enter the roll numbers: 1 2 3 4 5 6 7 8 9 10

Enter the marks in 5 subjects for 1: 1 1 1 1 1

Enter the marks in 5 subjects for 2: 2 2 2 2 2

Enter the marks in 5 subjects for 3: 3 3 3 3 3

Enter the marks in 5 subjects for 4: 4 4 4 4 4

Enter the marks in 5 subjects for 5: 5 5 5 5 5

Enter the marks in 5 subjects for 6: 6 6 6 6 6

Enter the marks in 5 subjects for 7: 7 7 7 7 7

Enter the marks in 5 subjects for 8: 8 8 8 8 8

Enter the marks in 5 subjects for 9: 9 9 9 9 9

Enter the marks in 5 subjects for 10: 0 0 0 0 0

Highest marks in subject 1 is received by 9

Second highest marks in subject 1 is received by 8

Highest marks in subject 2 is received by 9

Second highest marks in subject 2 is received by 8

Highest marks in subject 3 is received by 9

Second highest marks in subject 3 is received by 8

Highest marks in subject 4 is received by 9

Second highest marks in subject 4 is received by 8

Highest marks in subject 5 is received by 9

Second highest marks in subject 5 is received by 8

1. Write a C program which accepts a matrix and prints its transpose.

Code:

#include<stdio.h>

int main(){

printf("Enter the order of the matrix : (R C) ");

int m,n;

scanf("%d %d",&m,&n);

int a[m][n];

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

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

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

}

}

//Creating the transpose matrix

int b[n][m];

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

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

b[i][j]=a[j][i];

}

}

printf("The transpose is :\n");

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

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

printf("%d ",b[i][j]);

}

printf("\n");

}

return 0;

}

Output:

Enter the order of the matrix : (R C) 3 2

1 2

3 4

5 6

The transpose is :

1 3 5

2 4 6

1. Write a C program to replace a square matrix by its transpose without using a second matrix.

Code:

#include<stdio.h>

int main(){

printf("Enter the order of the matrix :");

int n;

scanf("%d",&n);

int a[n][n];

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

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

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

}

}

int temp,k=0;

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

for (int j=k;j<n;j++){

temp=a[i][j];

a[i][j]=a[j][i];

a[j][i]=temp;

}

k++;

}

printf("The transpose of the matrix is :\n");

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

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

printf("%d ",a[i][j]);

}

printf("\n");

}

return 0;

}

Output:

Enter the order of the matrix :3

1 2 3

3 4 5

7 8 9

The transpose of the matrix is :

1 3 7

2 4 8

3 5 9

1. Consider the following procedure:
2. Take as input any four-digit number, using at least two different digits. (Leading zeros are allowed.)
3. Arrange the digits in descending and then in ascending order to get two four-digit numbers, adding leading zeros if necessary.
4. Subtract the smaller number from the bigger number. Let the difference be the new four digit number.
5. Go back to step ii

The above process, known as Kaprekar's routine, will always reach a fixed point (Known as Kaprekar Constant). Write a C-Code to implement the algorithm given above and find out the constant number.

Note:

1. The fixed point is achieved when in two consecutive steps the same number is obtained
2. In C the binary arithmetic operation m%n gives the remainder when m is divided by n

Code:

#include<stdio.h>

int main(){

//inputting the number and putting in an array

int a[4]={0},ip,n,out,x;

printf("Enter any 4-digit number containing atleast 2 different digits : ");

scanf("%d",&ip);

out=ip;

while (out != x){

x=out;

n=x;

//Putting the number in an array

for (int i=0;i<4;i++){

a[i]=n%10;

n=n/10;

}

//All same digits

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

printf("!ERROR! Atleast 2 distict digits needed\n");

return 0;

}

//Sorting in ascending order

int min,min\_index,temp;

for (int i=0;i<3;i++){

min=a[i];

min\_index=i;

for (int j=i+1;j<4;j++){

if (a[j]<min){

min=a[j];

min\_index=j;

}

}

temp=a[i]+a[min\_index];

a[min\_index]=temp-a[min\_index];

a[i]=temp-a[i];

}

//Sorting in descending order in array b

int b[4]={0};

for (int i=0;i<4;i++){

b[i]=a[3-i];

}

//Creating the smaller number

int small=0;

for (int i=0;i<4;i++){

small=small\*10+a[i];

}

//Creating the bigger number

int big=0;

for (int i=0;i<4;i++){

big=big\*10+b[i];

}

out = big -small;

}

printf("%d\n",out);

return 0;

}

Output:

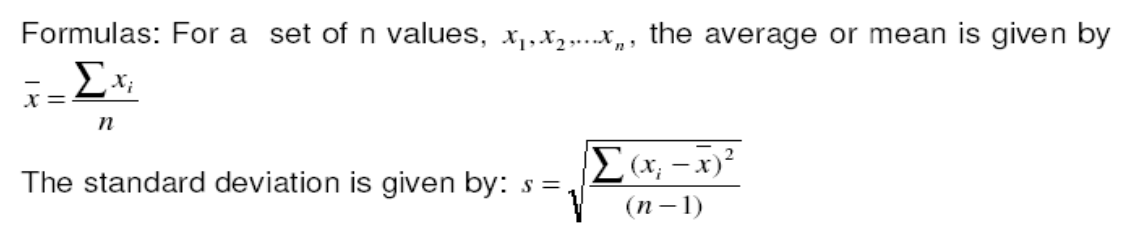
Enter any 4-digit number containing atleast 2 different digits : 1234

6174

Enter any 4-digit number containing atleast 2 different digits : 3241

6174

1. Write a program which takes some numbers and computes the standard deviation of them



Code:

#include<stdio.h>

#include<math.h>

int main(){

int n;

printf("Enter number of elements : ");

scanf("%d",&n);

float a[n];

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

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

}

//Calculating the mean value

float mean,sum=0;

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

sum+=a[i];

}

mean = sum/n;

//Calculating the standard deviation

sum=0;

float sd;

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

sum+= pow ((a[i]-mean),2);

}

sd = sqrt (sum/(n-1));

printf("Standard deviation is %f\n",sd);

return 0;

}

Output:

Enter number of elements : 5

1 3 5 7 9

Standard deviation is 3.162278