

|  |
| --- |
| C.P.N.M. LAB REPORT |
|  |
| ASSIGNMENT 4  BCSE FIRST YEAR FIRST SEMESTER  Authored by: SOHAM CHOWDHURY |



**CPNM LAB ASSIGNMENT REPORT**

BCSE FIRST YEAR FIRST SEMESTER 2021-2022

NAME-SOHAM CHOWDHURY

DEPARTMENT-COMPUTER SCIENCE AND ENGINEERING

SECTION-A3.

ROLL NO-002110501145.

# ASSIGNMENT 4

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

I RAN A LOOP FROM INDEX 0 TO LENGTH/2 FOR AN ARRAY AND SWAPPED THE 1ST AND LAST 2ND AND SECOND LAST AND SO ON USING THIRD VARIABLE.

Program:

#include<stdio.h>

#include<stdlib.h>

int\* reverse(int l,int arr[])

{

    int temp;

    for(int i=0;i<l/2;i++)

    {

        int temp=arr[i];

        arr[i]=arr[l-i-1];

        arr[l-i-1]=temp;

    }

    return (arr);

}

int main()

{

    int\* arr,length;

    int \*ptr;

    printf("enter the length of the array");

    scanf("%d",&length);

    arr=(int\*)calloc(length,sizeof(int));

    printf("enter the elements of the array:-\n");

    for(int i=0;i<length;i++)

    scanf("%d",&arr[i]);

    printf("the old array is :-\n");

    for(int i=0;i<length;i++)

    printf("%d",arr[i]);

    printf("\n");

    ptr=reverse(length,arr);

    printf("the reversed array is:-\n");

    for(int i=0;i<length;i++)

    printf("%d",ptr[i]);

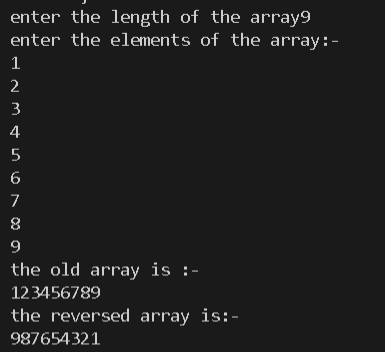
    free(arr);

    free(ptr);

    return 0;

}

Output:



2.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.

Using nested loop I find the repeating elements in the array and as soon as a duplicate is found I increase the counter and delete the duplicate element as well.

Program:

#include<stdio.h>

#include<stdlib.h>

int main()

{

    int \* arr=NULL;

    int n,d,c=0;

    printf("enter the number of elements=");

    scanf("%d",&n);

    arr=(int\*)calloc(n,sizeof(int));

    printf("enter the elements of the array");

    int l=n;

    for(int i=0;i<n;i++)

    {

        scanf("%d",&arr[i]);

    }

    for(int i=0;i<l-1;i++)

    {

        for(int j=i+1;j<l;j++)

        {

            if(arr[i]==arr[j])

            {

                c++;

                for(int k=j;k<l-1;k++)

                arr[k]=arr[k+1];

                l--;

                j--;

            }

        }

    }

    printf("the number of duplicate elements in the array = %d\n",c);

    printf("the new array is:-\n");

    for(int i=0;i<l;i++)

    printf("%d",arr[i]);

    int temp;

    for(int i=0;i<l/2;i++)

    {

        temp=arr[i];

        arr[i]=arr[l-i-1];

        arr[l-i-1]=temp;

    }

    printf("\nthe new reversed array is:-\n");

    for(int i=0;i<l;i++)

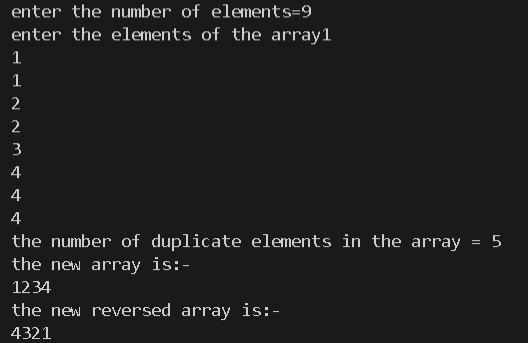
    printf("%d",arr[i]);

    free(arr);

    return 0;

}

Output:



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

Sum can be done by running nested loops for rows and columns respectively and summing the respective elements and storing it in a new matrix.

Same applies for diiference except now the difference is stored in the new array.

For product we multiply respective elements of row from 1st matrix and column of 2nd matrix then sum of products is stored in the new matrix.

Program:

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int main()

{

    int a[10][10];

    int b[10][10];

    int n,choice;

    int c[10][10];

    printf("enter the number of rows/columns of 1st & 2nd array=");

    scanf("%d",&n);

    printf("enter the 1st array=");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    printf("enter the 2nd array=");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            scanf("%d",&b[i][j]);

        }

    }

    printf("1 for sum , 2 for difference and 3 for product");

    printf("enter the choice=");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1:

        printf("the reslutant array is:-\n");

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

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

            }

        }

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

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

            }

            printf("\n");

        }

        break;

        case 2:

        printf("the reslutant array is:-\n");

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

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

            }

        }

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

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

            }

            printf("\n");

        }

        break;

        case 3:

        printf("the reslutant array is:-\n");

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

                int sum=0;

                for(int c=0;c<n;c++)

                {

                    sum+=a[i][c]\*b[c][j];

                }

                c[i][j]=sum;

            }

        }

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<n;j++)

            {

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

            }

            printf("\n");

        }

        break;

        default:

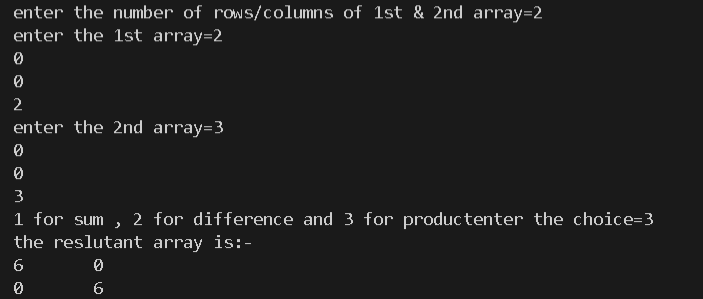
        printf("invalid choice");

    }

    return 0;

}

Output:



4.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.

Take integers in an array find maximum and minimum then find ther difference.

Program:

#include<stdio.h>

#include<stdlib.h>

int main()

{

    int l,max=0,min;

    int\* arr;

    printf("enter the length of the array=");

    scanf("%d",&l);

    arr=(int\*)calloc(l,sizeof(int));

    printf("enter the elements of the array:-\n");

    for(int i=0;i<l;i++)

    scanf("%d",&arr[i]);

    min=arr[0];

    for(int i=0;i<l;i++)

    {

        if(max<arr[i])

        max=arr[i];

        if(min>arr[i])

        min=arr[i];

    }

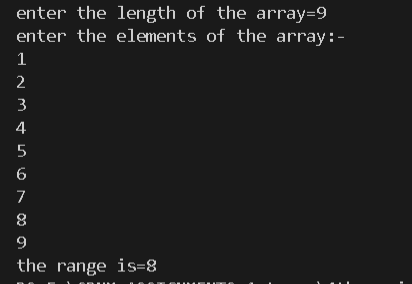
    printf("the range is=%d",max-min);

    free(arr);

    return 0;

}

Output:



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

Here I took 10 numbers from user and applied selection sort to sort them which has a time complexity of O(n\*n) for worst case.

Program:

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

    int \*arr;

    int l;

    printf("enter the length of the array=");

    scanf("%d",&l);

    arr=(int\*)calloc(l,sizeof(int));

    printf("enter the elements of the array:-\n");

    for(int i=0;i<l;i++)

    scanf("%d",&arr[i]);

    int temp;

    for(int i=0;i<l-1;i++)

    {

        for(int i=0;i<l-1;i++)

        {

            if(arr[i]>arr[i+1])

            {

                temp=arr[i+1];

                arr[i+1]=arr[i];

                arr[i]=temp;

            }

        }

    }

    printf("the new array is:-\n");

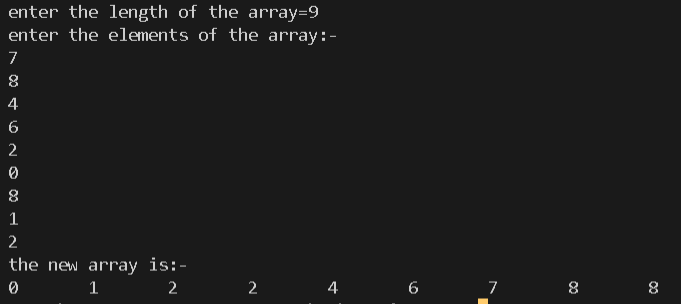
    for(int i=0;i<l;i++)

    printf("%d\t",arr[i]);

    return 0;

}

Output:



6.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.

Take marks of students in five arrays for five subjects and name of the students in a 2d array then find highest and second highest for each array .

Program:

#include<stdio.h>

#include<string.h>

int main()

{

    char name[10][15];

    int roll[10];

    int a[10];

    int b[10];

    int c[10];

    int d[10];

    int e[10];

    for(int i=0;i<10;i++)

    {

        printf("enter credentials of student %d:= i.e, name roll no and marks in 5 subjects\n",i+1);

        gets(name[i]);

        fflush(stdin);

        scanf("%d",&roll[i]);

        fflush(stdin);

        scanf("%d",&a[i]);

        fflush(stdin);

        scanf("%d",&b[i]);

        fflush(stdin);

        scanf("%d",&c[i]);

        fflush(stdin);

        scanf("%d",&d[i]);

        fflush(stdin);

        scanf("%d",&e[i]);

        fflush(stdin);

    }

    int n1,n2,max1=0,max2=0;

    for(int i=0;i<10;i++)

    {

        if(max1<a[i])

        {

            max1=a[i];

            n1=i;

        }

    }

    printf("the student who got highest marks in subject A is %s roll no-%d marks -%d\n",name[n1],roll[n1],max1);

    for(int i=0;i<10;i++)

    {

        if((a[i]>max2)&&(a[i]<max1))

        {

            max2=a[i];

            n2=i;

        }

    }

    printf("the student who got 2nd highest marks in subject A is %s roll no-%d marks -%d\n",name[n2],roll[n2],max2);

    max1=0;

    max2=0;

    for(int i=0;i<10;i++)

    {

        if(max1<b[i])

        {

            max1=b[i];

            n1=i;

        }

    }

    printf("the student who got highest marks in subject B is %s roll no-%d marks -%d\n",name[n1],roll[n1],max1);

    for(int i=0;i<10;i++)

    {

        if((b[i]>max2)&&(b[i]<max1))

        {

            max2=b[i];

            n2=i;

        }

    }

    printf("the student who got 2nd highest marks in subject B is %s roll no-%d marks -%d\n",name[n2],roll[n2],max2);

    max1=0;

    max2=0;

    for(int i=0;i<10;i++)

    {

        if(max1<c[i])

        {

            max1=c[i];

            n1=i;

        }

    }

    printf("the student who got highest marks in subject C is %s roll no-%d marks -%d\n",name[n1],roll[n1],max1);

    for(int i=0;i<10;i++)

    {

        if((c[i]>max2)&&(c[i]<max1))

        {

            max2=c[i];

            n2=i;

        }

    }

    printf("the student who got 2nd highest marks in subject C is %s roll no-%d marks -%d\n",name[n2],roll[n2],max2);

    max1=0;

    max2=0;

    for(int i=0;i<10;i++)

    {

        if(max1<d[i])

        {

            max1=d[i];

            n1=i;

        }

    }

    printf("the student who got highest marks in subject D is %s roll no-%d marks -%d\n",name[n1],roll[n1],max1);

    for(int i=0;i<10;i++)

    {

        if((d[i]>max2)&&(d[i]<max1))

        {

            max2=d[i];

            n2=i;

        }

    }

    printf("the student who got 2nd highest marks in subject D is %s roll no-%d marks -%d\n",name[n2],roll[n2],max2);

    max1=0;

    max2=0;

    for(int i=0;i<10;i++)

    {

        if(max1<e[i])

        {

            max1=e[i];

            n1=i;

        }

    }

    printf("the student who got highest marks in subject E is %s roll no-%d marks -%d\n",name[n1],roll[n1],max1);

    for(int i=0;i<10;i++)

    {

        if((e[i]>max2)&&(e[i]<max1))

        {

            max2=e[i];

            n2=i;

        }

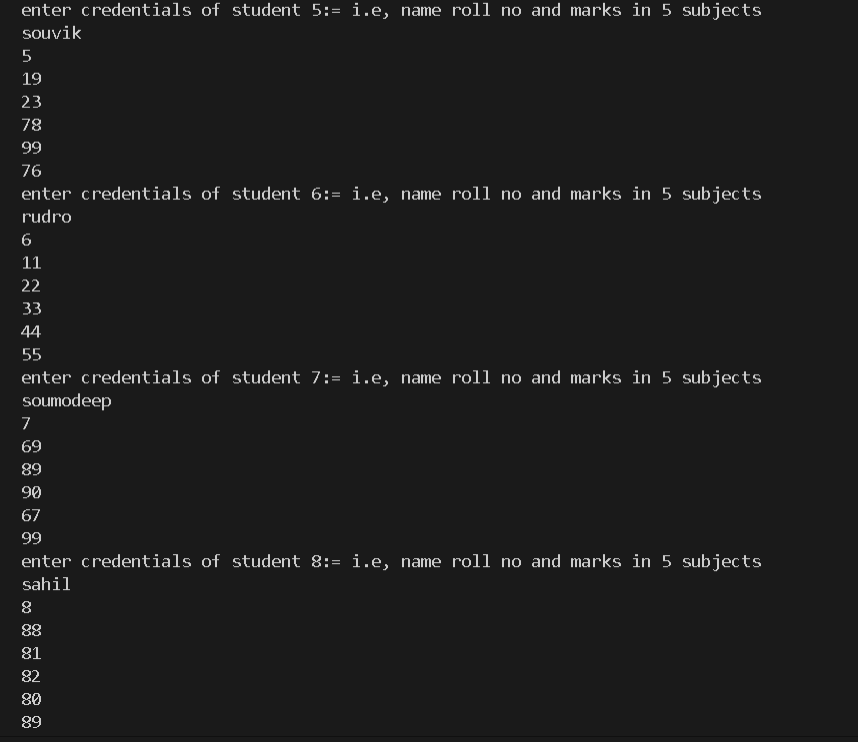
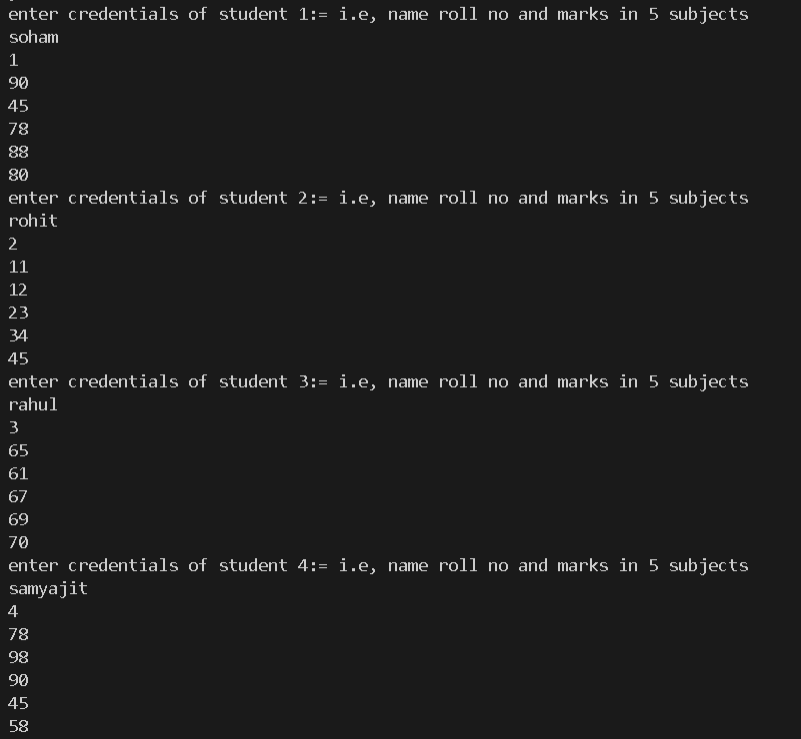
    }

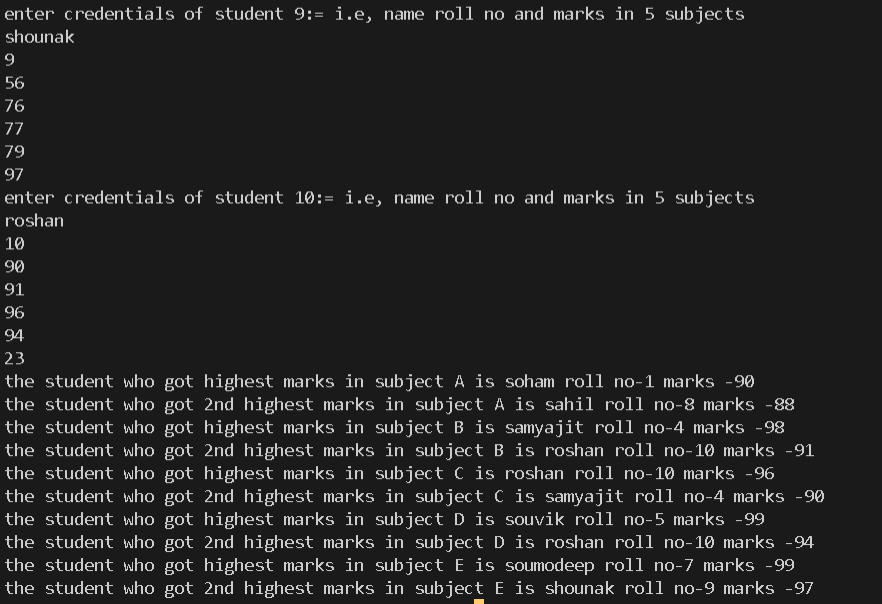
    printf("the student who got 2nd highest marks in subject E is %s roll no-%d marks -%d\n",(name[n2]),roll[n2],max2);

    return 0;

}

Output:





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

Take matrix from user make another matrix and put ith row jth column of first matrix to jth row ith column of the new matrix.

Program: #include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

    int arr[10][10];

    int arr1[10][10];

    int n,m;

    printf("enter the rows & columns of the array \n");

    scanf("%d%d",&n,&m);

    printf("enter the elements of the array:-\n");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)

        scanf("%d",&arr[i][j]);

    }

    printf("the given array is:-\n");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)

        printf("%d\t",arr[i][j]);

        printf("\n");

    }

    for(int j=0;j<m;j++)

    {

        for(int i=0;i<n;i++)

        {

            arr1[i][j]=arr[i][j];

        }

    }

    printf("the new array transposed is:-\n");

    for(int j=0;j<m;j++)

    {

        for(int i=0;i<n;i++)

        printf("%d\t",arr1[i][j]);

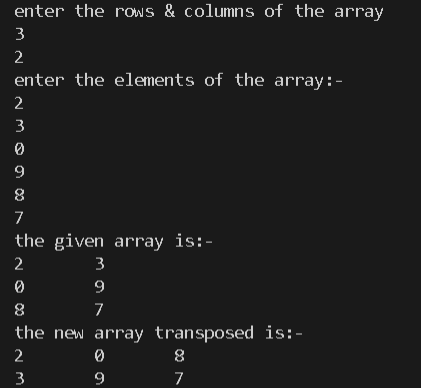
        printf("\n");

    }

    return 0;

}

Output:



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

Take the initial matrix from the user and swap ith row jth column of first matrix to jth row ith column of the same matrix

Program: #include<stdio.h>

#include<math.h>

int main()

{

    int arr[10][10];

    int n,temp;

    printf("enter the length of the square matrix=");

    scanf("%d",&n);

    printf("enter the matrix:-\n");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            scanf("%d",&arr[i][j]);

        }

    }

    printf("the old matrix is:-\n");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            printf("%d\t",arr[i][j]);

        }

        printf("\n");

    }

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            if(j>=i)

            {

                temp=arr[i][j];

                arr[i][j]=arr[j][i];

                arr[j][i]=temp;

            }

        }

    }

    printf("the new transposed matrix is matrix is:-\n");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            printf("%d\t",arr[i][j]);

        }

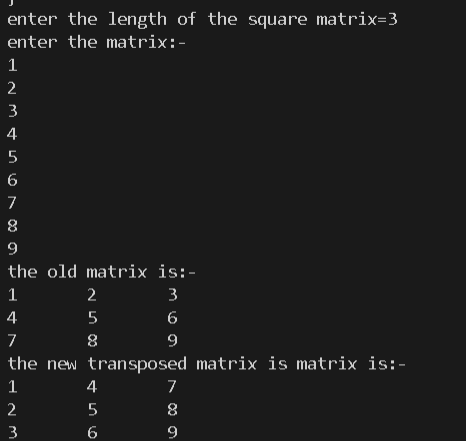
        printf("\n");

    }

    return 0;

}

Output:



9.Consider the following procedure: i. Take as input any four-digit number, using at least two different digits. (Leading zeros are allowed.) ii. Arrange the digits in descending and then in ascending order to get two four-digit numbers, adding leading zeros if necessary. iii. Subtract the smaller number from the bigger number. Let the difference be the new four digit number. iv. 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. Also create an output file ‘output.dat’ in the working folder and write the following with appropriate format for each step of iteration: The 4-digit number, the larger number, the smaller number and the difference of the larger and the smaller number. Note: A. The fixed point is achieved when in two consecutive steps the same number is obtained B. In C the binary arithmetic operation m%n gives the remainder when m is divided by n

Took the number from the user sorted it in ascending and descending order using selection sorting and reversing respectively,found the difference between ascending no and descending number stored it in an array and went on till the current element of the array is equal to the previous element of the array.

Program:

#include<stdio.h>

#include<math.h>

int main()

{

    int a[4],b[4],diff[100];

    int k=0,f=0;

    printf("enter the number (digit wise)=");

    for(int i=0;i<4;i++)

    scanf("%d",&a[i]);

    while(1)

    {

        int temp,num1=0,num2=0;

        for(int i=0;i<3;i++)

        {

            for(int i=0;i<3;i++)

            {

                if(a[i]>a[i+1])

                {

                    temp=a[i];

                    a[i]=a[i+1];

                    a[i+1]=temp;

                }

            }

        }

        for(int i=0;i<4;i++)

        {

            b[i]=a[4-i-1];

        }

        for(int i=0;i<4;i++)

        {

            num1=num1\*10+a[i];

        }

        for(int i=0;i<4;i++)

        {

            num2=num2\*10+b[i];

        }

        if(num2>num1)

        k=num2-num1;

        else

        k=num1-num2;

        diff[f++]=k;

        if((f>=2)&&(diff[f-1]==diff[f-2]))

        {

            k=diff[f-1];

            break;

        }

        int c=0;

        for(int i=k;i>0;i/=10)

        c++;

        int l=3;

        if(c==4)

        {

            for(int i=k;i>0;i/=10)

            {

                a[l--]=i%10;

            }

            l=3;

        }

        else if(c<4)

        {

            int g=3;

            for(int i=k;i>0;i=i/10)

            {

                if((g<=l-c)&&(g>=0))

                {

                    a[g--]=0;

                }

                else

                {

                    a[g--]=i%10;

                }

            }

        }

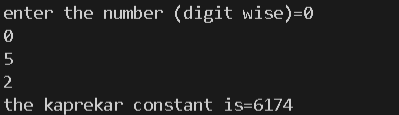
    }

    printf("the kaprekar constant is=%d",k);

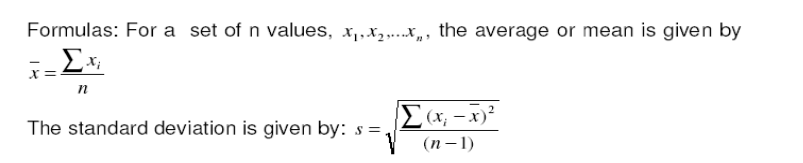
    return 0;

}

Output:



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



Found the mean by traversing the user defined array and getting the sum of each element of the array by traversal and devided it with length to find mean.

Then standard devitation by again traversing the array and applieying the formula given above in the form of a mathematical expression.

Program:

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

    int \* arr;

    int l;

    double mean,sum=0,sum1=0,s;

    printf("enter the length of the array=");

    scanf("%d",&l);

    arr=(int\*)malloc(l\*sizeof(int));

    for(int i=0;i<l;i++)

    scanf("%d",&arr[i]);

    for(int i=0;i<l;i++)

    sum+=arr[i];

    mean=sum/l;

    printf("the elements of the array is:-\n");

    for(int i=0;i<l;i++)

    printf("%d\t",arr[i]);

    printf("\nmean=%f",mean);

    for(int i=0;i<l;i++)

    {

        sum1+=pow(arr[i]-mean,2);

    }

    s=sqrt(sum1/(l-1));

    printf("\nthe standard deviation is=%0.2f",s);

    free(arr);

    return 0;

}

Output:

