**Rajeev Gandhi Memorial College of Engineering and Technology**

**Autonomous**

**MASTER OF COMPUTER APPLICATIONS**

II YEAR, I-SEM (MCA)

**(F0081203) DATA WAEHOUSING AND DATA MINING LAB**

Exercise No: 1A

**AIM: Write a C program to perform data cleaning techniques using smoothing by bin means.**

Data Pre-processing :

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behavior’s or trends, and is likely to contain many errors. Data pre-processing is a proven method of resolving such issues. Data pre- processing prepares raw data for further processing. Data preprocessing is used database-driven applications such as customer relationship management and rule- based applications (like neural networks).

Data Smoothing :

Data smoothing is done by using an algorithm to remove noise from a dataset. This allows important patterns to stand out. Data smoothing can be used to help predict trends, such as those found in securities prices. The idea behind data smoothing is that it can identify simplified changes in order to help predict different trends and patterns. It acts as an aid for statisticians or traders who need to look at a lot of data that can often be complicated to digest to find patterns they would not otherwise see.

Smoothing by Bin Means :

In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

Example:

Perform data cleaning technique using smoothing by BIN MEANS on elements

4, 8, 15, 21, 21, 24, 25, 34, 28 whose bin size is 3.

Sort the elements and form a matrix using bin size.

4 8 15

21 21 24

25 28 34

**Calculate mean value for each bin. Bin 1: 4 8 15**

**Bin 1 Mean: sum of elements of bin 1/bin size**

= (4+8+15)/3

= 9

Bin 2: 21 21 24

**Bin 2 Mean: sum of elements of bin 2/bin size**

= (21+21+24)/3

= 22

Bin 3: 25 28 34

**Bin 3 Mean: sum of elements of bin 3/bin size**

= (25+28+34)/3

= 29

Mean Values:

9 9 9

22 22 22

29 29 29

Program:

#include<stdio.h>

void main()

{

int i, k, n, bin, a [50], temp, sum; float mean;

printf("Enter no of bins:"); scanf("%d",&bin);

printf("Enter no of values to be enter in each bin:"); scanf("%d",&n);

printf("Enter elements\n"); for(i=0;i<n\*bin;i++)

{

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

}

printf("The given elements are\n"); for(i=0;i<n\*bin;i++)

{

for(k=0;k<n\*bin;k++)

{

if(a[i]<a[k])

{

temp=a[i]; a[i]=a[k]; a[k]=temp;

}

}

}

for(i=0;i<bin;i++)

{

for(k=0;k<n;k++)

{

printf("%2d\t",a[k+i\*n]);

}

printf("\n");

}

printf("The mean values are\n"); for(i=0;i<bin;i++)

{

mean=0,sum=0; for(k=0;k<n;k++)

{

sum=sum+a[k+i\*n];

}

mean=(float)sum/n; for(k=0;k<n;k++)

{

a[k+i\*n]=mean; printf("%f\t",mean);

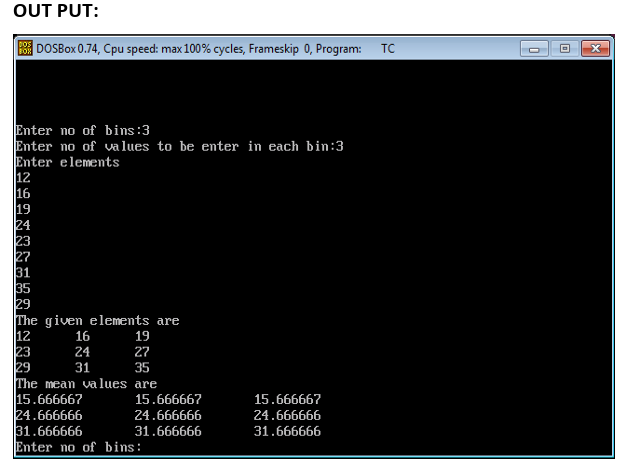
}

printf("\n");

}

}

OUT PUT:



**Exercise No:1B**

**AIM: Write a C program to perform data cleaning techniques using smoothing by bin medians.**

Smoothing by Bin Medians:

In Smoothing by Bin Medians, each value in a bin is replaced by the median value of the bin.

Example:

Perform data cleaning technique using smoothing by BIN MEDIANS on elements 4, 8, 15, 21, 21, 24, 25, 34, 28 whose bin size is 3.

Sort the elements and form a matrix using bin size.

4 8 15

21 21 24

25 28 34

**Calculate median value for each bin.**

Bin 1: 4 8 15

**Bin 1 Median: Middle value of the bin = 8**

Bin 2: 21 21 24

**Bin 2 Median: Middle value of the bin = 21**

Bin 3: 25 28 34

**Bin 3Median: Middle value of the bin = 28**

Median Values:

8 8 8

21 21 21

28 28 28

**Program:**

#include<stdio.h>

void main()

{

int i,k,n,bin,a[50],temp,sum;

float median;

printf("Enter no of bins:");

scanf("%d",&bin);

printf("Enter no of values to be enter in each bin:");

scanf("%d",&n);

printf("Enter elements\n");

for(i=0;i<n\*bin;i++)

{

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

}

printf("The given elements are\n");

for(i=0;i<n\*bin;i++)

{

for(k=0;k<n\*bin;k++)

{

if(a[i]<a[k])

{

temp=a[i];

a[i]=a[k];

a[k]=temp;

}

}

}

for(i=0;i<bin;i++)

{

for(k=0;k<n;k++)

{

printf("%2d\t",a[k+i\*n]);

}

printf("\n");

}

printf("The median values are\n");

for(i=0;i<bin;i++)

{

if(n%2==0)

{

median=(float)(a[n/2-1+i\*n]+a[n/2+i\*n])/2;

}

else

{

median=a[n/2+i\*n];

}

for(k=0;k<n;k++)

{

a[k+i\*n]=median;

printf("%f\t",median);

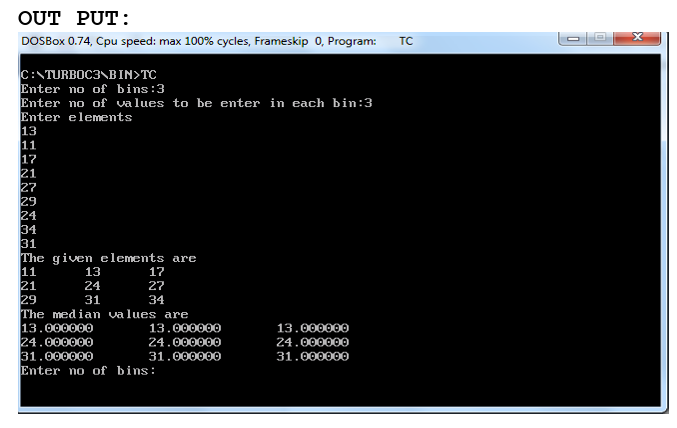
}

printf("\n");

}

}

OUT PUT:



**Exercise No:1C**

**AIM: Write a C program to perform data cleaning techniques using smoothing by bin boundaries.**

Smoothing by Bin Boundaries:

In Smoothing by Bin Boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries.

Example:

Perform data cleaning technique using smoothing by BIN BOUNDARIES on elements 4, 8, 15, 21, 21, 24, 25, 34, 28 whose bin size is 3.

Sort the elements and form a matrix using bin size.

4 8 15

21 21 24

25 28 34

Calculate boundary value for each element in the bins by finding difference of that element with lower and upper boundary values. The boundary value which gives minimum difference will be replaced in that element place in the bin.

Bin 1: 4 8 15

Bin 1:

Min=4 and Max=15

For value 4, (4-4) < (15-4) so replace 4 with lower boundary value 4.

For value 8, (8-4) < (15-8) so replace 8 with lower boundary value 4.

For value 15, (15-4) > (15-15) so replace 15 with upper boundary value 15.

Bin 2: 21 21 24

Bin 2:

Min=21 and Max=24

For value 21, (21-21) < (24-21) so replace 21 with lower boundary value 21.

For value 21, (21-21) < (24-21) so replace 21 with lower boundary value 21.

For value 24, (24-21) > (24-24) so replace 24 with upper boundary value 24.

Bin 3: 25 28 34

Bin 3:

Min=25 and Max=34

For value 25, (25-25) < (34-25) so replace 25 with lower boundary value 25.

For value 28, (28-25) < (34-28) so replace 28 with lower boundary value 25.

For value 34, (34-25) > (34-34) so replace 34 with upper boundary value 34. Bin

Boundaries:

4 4 15

21 21 24

25 25 34

Program:

#include<stdio.h>

void main()

{

int i,j,n,nb,a[50],temp,sum,mid,b[10][10];

printf("Enter no of bins:");

scanf("%d",&n);

printf("Enter values each bin:");

scanf("%d",&nb);

printf("Enter values\n");

for(i=0;i<n\*nb;i++)

{

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

}

printf("The given values\n");

for(i=0;i<n\*nb;i++)

{

for(j=0;j<n\*nb;j++)

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

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

{

for(j=0;j<nb;j++)

{

printf("%2d\t",a[j+i\*nb]);

b[i][j]=a[j+i\*nb];

}

printf("\n");

}

printf("boundaries are\n");

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

{

sum=b[i][0]+b[i][nb-1];

mid=sum/2;

for(j=0;j<nb;j++)

{

if(b[i][j]<=mid)

b[i][j]=b[i][0];

else

b[i][j]=b[i][n-1];

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

}

printf("\n");

}

}

OUT PUT:

