**Question 01:**

#include <stdio.h>

#include <conio.h>

int main()

{

int rows, columns, stride, filter\_rows, filter\_columns, i = 0, j = 0, k = 0, l = 0, max = 0;

int \*ptr\_rows=&rows, \*ptr\_columns=&columns, \*ptr\_stride=&stride, \*ptr\_filter\_rows=&filter\_rows, \*ptr\_filter\_columns=&filter\_columns, \*ptr\_max =&max;

printf("Enter the number of rows you want in the matrix\n");

scanf("%d", &\*ptr\_rows);

printf("Enter the number of columns you want in the matrix\n");

scanf("%d", &\*ptr\_columns);

printf("Enter the number of rows in the filter\n");

scanf("%d", &\*ptr\_filter\_rows);

printf("Enter the number of columns in the filter\n");

scanf("%d", &\*ptr\_filter\_columns);

printf("Enter the value of the stride\n");

scanf("%d", &\*ptr\_stride);

int matrix[rows][columns];

printf("Enter the values of the matrix\n");

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

{

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

{

printf("Enter the value for the (%d,%d) position in the matrix\n", i + 1, j + 1);

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

}

}

printf("The matrix you entered is as follows:\n");

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

{

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

{

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

}

printf("\n");

}

int pooled\_rows = ((\*ptr\_rows - \*ptr\_filter\_rows) / (\*ptr\_stride)) + 1;

int pooled\_columns = ((\*ptr\_columns - \*ptr\_filter\_columns) / (\*ptr\_stride)) + 1;

struct pool

{

int pooled\_rows;

int pooled\_columns;

int pooled\_matrix[pooled\_rows][pooled\_columns];

};

struct pool pool,\*ptr\_pool=&pool;

for (k = 0; k < pooled\_rows; k++)

{

for (l = 0; l < pooled\_columns; l++)

{

max = -999;

for (i = k \* stride; i < filter\_rows + k \* stride; i++)

{

for (j = l \* stride; j < filter\_columns + l \* stride; j++)

{

if (matrix[i][j] > \*ptr\_max)

{

\*ptr\_max = matrix[i][j];

}

}

}

ptr\_pool->pooled\_matrix[k][l]= \*ptr\_max;

}

}

printf("The max pooled matrix with a stride of %d from the given matrix is:\n",stride);

for (i = 0; i < pooled\_rows; i++)

{

for (j = 0; j < pooled\_columns; j++)

{

printf("%d ", ptr\_pool->pooled\_matrix[i][j]);

}

printf("\n");

}

}

**Question 02:**

#include <stdio.h>

#include <math.h>

void main()

{

int i, n;

printf("Enter the number of element you want in X and Y arrays\n");

scanf("%d", &n);

struct x\_and\_y

{

float x[n];

float y[n];

};

struct x\_and\_y sample,\*ptr\_sample=&sample;

float r = 0, sum\_x = 0, sum\_y = 0, mean\_x = 0, mean\_y = 0, numerator = 0, denominator = 0;

float \*ptr\_r = &r, \*ptr\_sum\_x = &sum\_x, \*ptr\_sum\_y = &sum\_y, \*ptr\_mean\_x = &mean\_x, \*ptr\_mean\_y = &mean\_y, \*ptr\_numerator = &numerator, \*ptr\_denominator = &denominator;

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

{

printf("Enter the value for %d position of x\n", i + 1);

scanf("%f", &ptr\_sample->x[i]);

\*ptr\_sum\_x += ptr\_sample->x[i];

}

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

{

printf("Enter the value for %d position of y\n", i + 1);

scanf("%f", &ptr\_sample->y[i]);

\*ptr\_sum\_y += ptr\_sample->y[i];

}

\*ptr\_mean\_x = \*ptr\_sum\_x / n;

\*ptr\_mean\_y = \*ptr\_sum\_y / n;

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

{

\*ptr\_numerator += (ptr\_sample->x[i] - \*ptr\_mean\_x) \* (ptr\_sample->y[i] - \*ptr\_mean\_y);

\*ptr\_denominator += sqrt(pow((ptr\_sample->x[i] - \*ptr\_mean\_x), 2) \* pow((ptr\_sample->y[i] - \*ptr\_mean\_y), 2));

if (\*ptr\_denominator==0)

{

printf("MATH ERROR!\n");

return;

}

}

\*ptr\_r = \*ptr\_numerator / \*ptr\_denominator;

printf("The value of R(Pearson coefficient of correlation) is %.3f ", \*ptr\_r);

//getch();

}

**Question 03:**

#include <stdio.h>

#include <math.h>

int main()

{

int i, n;

printf("Enter the number of points you want in P and Q arrays\n");

scanf("%d", &n);

struct P\_and\_Q

{

float P[n];

float Q[n];

};

struct P\_and\_Q points;

struct P\_and\_Q \*ptr\_points = &points;

float d = 0, a;

float \*ptr\_d = &d;

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

{

printf("Enter the value for %d position of x\n", i + 1);

scanf("%f", &ptr\_points->P[i]);

}

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

{

printf("Enter the value for %d position of y\n", i + 1);

scanf("%f", &ptr\_points->Q[i]);

}

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

{

\*ptr\_d += pow((ptr\_points->Q[i] - ptr\_points->P[i]), 2);

}

\*ptr\_d = sqrt(\*ptr\_d);

printf("The value of Euclidian distance of the given points is %.3f\n", \*ptr\_d);

getchar();

}

**Question 04:**

#include <stdio.h>

#include <math.h>

float func\_StanDev(float \*arr,int \*size)

{

int i=0;float numerator=0,sum=0,avg=0;

for ( i = 0; i < \*size; i++)

{

sum+=\*(arr+i);

}

//printf("The sum is %.2f\n",sum);

avg=sum/(\*size);

for ( i = 0; i < \*size; i++)

{

numerator+=pow((\*(arr+i)-avg),2);

}

return sqrt(numerator/(\*size));

}

int main()

{

float standDev;

float arr[8] = {10,12,23,23,16,23,21,16};

int n=sizeof(arr)/sizeof(float),\*ptr\_n=&n;

int i;

printf("The data you provided is:\n");

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

{

printf("%.2f ",arr[i]);

}

printf("\n");

float \*ptr\_arr=&arr[0];

printf("The Standard Deviation of the given data is %.3f.",func\_StanDev(ptr\_arr,ptr\_n));

}