Assignment 5

20K-0157 Section D

**Task1**

#include<stdlib.h>

#include<stdio.h>

struct matrix {

int\*\* array;

int rows;

int colums;

};

int pooledvalues(int fin\_arr, int row, int col,int\*\* array ) {

int max = -9999,i,j;

for (i = row; i < row + fin\_arr; i++) {

for (j = col; j < col + fin\_arr; j++) {

if (array[i][j] > max)

max = array[i][j];

}

}

return max;

}

void PrintFunc(int size, int\*\* array) {

int i,j;

for (i = 0; i < size; i++) {

for (j = 0; j < size; j++) {

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

}

printf("\n");

}

}

int main() {

struct matrix mat1;

struct matrix poolmat;

int arraysize, stride, fin\_arr,i,j;

printf("How big is the square matrix ?");

scanf("%d", &arraysize);

printf("Enter Filter ");

scanf("%d", &fin\_arr);

printf("Print Stride is 2 ");

stride=2;

mat1.array = (int\*\*)malloc(arraysize \* sizeof(int));

for (i = 0; i < arraysize; i++) {

mat1.array[i] = (int\*)malloc(arraysize \* sizeof(int));

for (j = 0; j < arraysize; j++) {

printf("Enter value ");

scanf("%d", &mat1.array[i][j]);

}

}

poolmat.array = (int\*\*)malloc(fin\_arr \* sizeof(int));

for (i = 0; i < arraysize; i++) {

poolmat.array[i] = (int\*)malloc(fin\_arr \* sizeof(int));

}

PrintFunc(arraysize, mat1.array);

printf("\n");

int fin\_col= 0,fin\_row=0,temp;

for (i = 0; i < arraysize - fin\_arr + 1; i += stride) {

for (j = 0; j < arraysize - fin\_arr + 1; j += stride) {

temp =pooledvalues( fin\_arr, i, j, mat1.array);

poolmat.array[fin\_col][fin\_row] = temp;

fin\_row++;

}

fin\_row = 0;

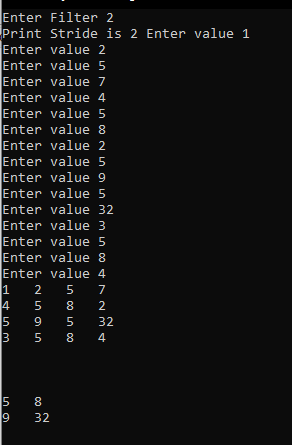
printf("\n");

fin\_col++;

}

PrintFunc(fin\_arr, poolmat.array);

}



**Task2**

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

int avereage(int array[], int size) {

int sum = 0,i;

for (i = 0; i < size; i++) {

sum += array[i];

}

return sum / size;

}

void PCal(float\* r, int\* x, int\* y, float xavg, float yavg, int size) {

float xtot = 0, ytot = 0,a=0,b=0;

int i;

for (i = 0; i < size; i++) {

xtot = xtot+(x[i] - xavg);

ytot = ytot+ (y[i] - yavg);

}

a = xtot \* ytot;

b = sqrt(pow(xtot, 2) \* pow(ytot, 2));

\*r = (float)(a / b);

}

int main() {

int x[9] = { 1, 3, 5, 7, 9, 11, 12, 14, 16 };

int y[9] = { 2, 4, 6, 8, 10, 12, 13, 15, 17 };

float xavg = avereage(x, sizeof(x) / sizeof(x[0]));

float yavg = avereage(y, sizeof(y) / sizeof(y[0]));

float r1;

PCal(&r1, x, y, xavg, yavg, sizeof(x) / sizeof(x[0]));

printf("Pearson Coeff. of the data stored is %.4f", r1);

}



**Task3**

#include <stdio.h>

#include <math.h>

void formula(float\* distance, int\* x, int\* y, int size) {

float sum1 = 0;

int i;

for ( i = 0; i < size; i++) {

sum1 += pow((y[i] - x[i]),2);

}

\*distance = sqrt(sum1);

}

int main()

{

float distance;

int p [9] = { 1, 3, 5, 7, 9, 11, 12, 14, 16 };

int q[9] = { 2, 4, 6, 8, 10, 12, 13, 15, 17 };

int size=sizeof(p ) / sizeof(p [0]);

formula(&distance, p , q, size);

printf("Euclidian distance of the stored data is %.3f", distance);

}



**Task4**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

void func(int \*sum,float \*sd,float \*avg,int array[])

{

int n=5,i,j;

float submission=0,a=0;

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

{

\*sum=\*sum+array[i];

}

\*avg=\*sum/n;

for(j=0;j<=n-1;j++)

{

a=((array[j]-\*avg)\*(array[j]-\*avg));

submission=submission+a;

}

\*sd=sqrt(submission/n);

}

int main(int argc, char \*argv[]) {

int sum=0;

float sd=0,avg=0;

int arr[8],k;

printf("Enter values\n");

for(k=0;k<=7;k++)

{

scanf("%d",&arr[k]);

}

func(&sum,&sd,&avg,arr);

printf("SD = %.3f",sd);

return 0;

}

