**SOURCE CODE (C++):**#include <iostream>

#include <iomanip>

#include <cmath>

#define MAX\_SIZE 20

#define LIN 0

#define EXP 1

class CurveFit

{

public:

CurveFit() ;

void getDataset() ;

void displayDataset() ;

void fitToLinear() ;

void fitToExponential() ;

private:

typedef struct { float x, y ;} pair ;

int num ;

float sumX, sumY, sumXY, sumX2, sumlogY, sumXlogY ;

pair dataset[MAX\_SIZE], getAb ;

void calcSum(){

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

sumX += dataset[i].x ;

sumY += dataset[i].y ;

sumX2 += dataset[i].x \* dataset[i].x ;

sumXY += dataset[i].x \* dataset[i].y ;

sumlogY += log(dataset[i].y) ;

sumXlogY += log(dataset[i].y) \* dataset[i].x ;

}

}

pair solveCramer2D(float A00, float A01, float A10, float A11, float B0, float B1){

// Cramer's Rule

pair xy ;

float det, detX, detY ;

det = (A00\*A11 - A01\*A10) ;

detX = (B0\*A11 - A01\*B1) ;

detY = (A00\*B1 - B0\*A10) ;

xy.x = detX/det ;

xy.y= detY/det ;

return xy ;

}

};

CurveFit::CurveFit(){ sumX = sumY = sumXY = sumX2 = sumlogY = sumXlogY = 0 ;}

void CurveFit::getDataset(){

std::cout << "!!!!! CURVE FITTING !!!!!" << std::endl ;

std::cout << "Input the number of points in the dataset to fit: " ;

std::cin >> num ;

if(num > MAX\_SIZE){

std::cout << "Element limit exceeded !" << std::endl ;

}

else{

std::cout << "Dataset -> " << std::endl ;

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

std::cout << i+1 << ". " ;

std::cout << "X: " ;

std::cin >> dataset[i].x ;

std::cout << " Y: " ;

std::cin >> dataset[i].y ;

}

}

calcSum() ;

}

void CurveFit::displayDataset(){

std::cout << std::setw(10) << "X" << std::setw(10) << "Y" << std::endl ;

std::cout << "------------------------------" << std::endl ;

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

std::cout << std::setw(10) << dataset[i].x << std::setw(10) << dataset[i].y << std::endl ;

}

}

void CurveFit::fitToLinear(){

getAb = solveCramer2D(num, sumX, sumX, sumX2, sumY, sumXY) ;

std::cout << "A: " << getAb.x << std::endl ;

std::cout << "B: " << getAb.y << std::endl ;

std::cout << "Linear Fit -> Y = " << getAb.x << " + " << getAb.y << " X "<< std::endl ;

}

void CurveFit::fitToExponential(){

getAb = solveCramer2D(num, sumX, sumX, sumX2, sumlogY, sumXlogY) ;

std::cout << "A: " << exp(getAb.x) << std::endl ;

std::cout << "B: " << getAb.y << std::endl ;

std::cout << "Exponential Fit -> Y = " << exp(getAb.x) << " exp( " << getAb.y << " X ) "<< std::endl ;   
}

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

CurveFit \*cf = nullptr ;

cf = new CurveFit() ;

cf->getDataset() ;

int select ;

std::cout << "Fit to ? \n 1. Linear\t 2. Exponential\n => " ;

std::cin >> select ;

cf->displayDataset() ;

std::cout << "Solution: " << std::endl ;

switch(select-1){

case LIN : cf->fitToLinear() ; break ;

case EXP : cf->fitToExponential() ; break ;

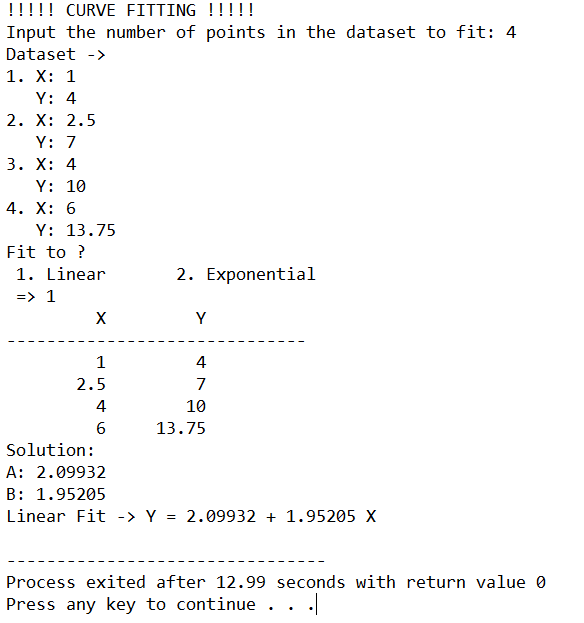
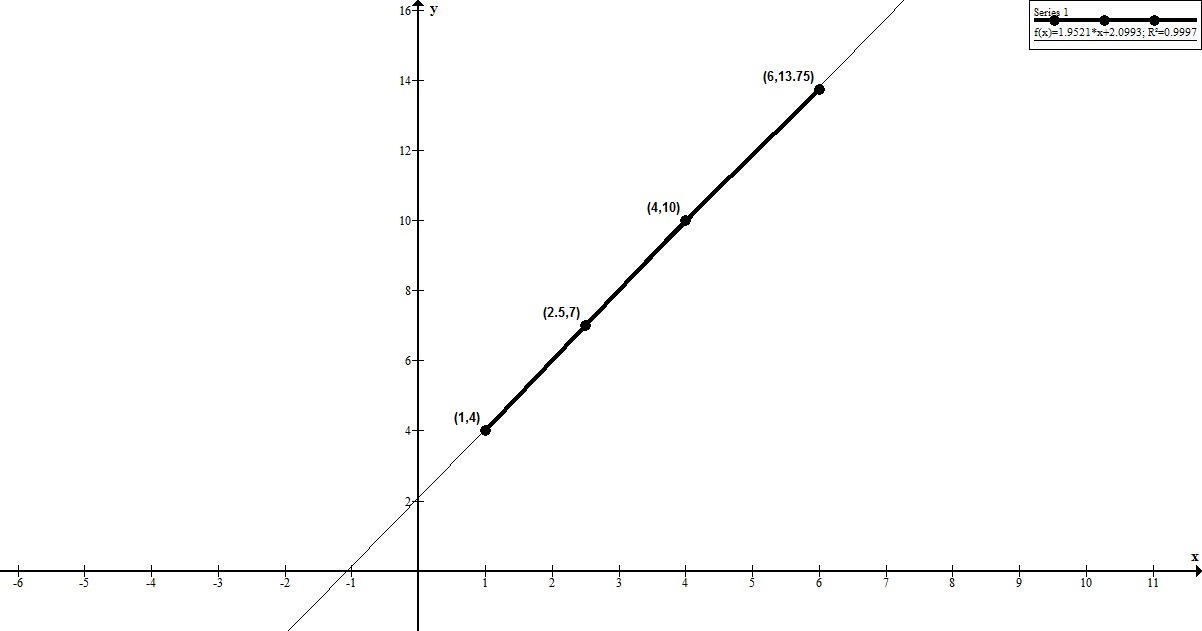
default : break ;

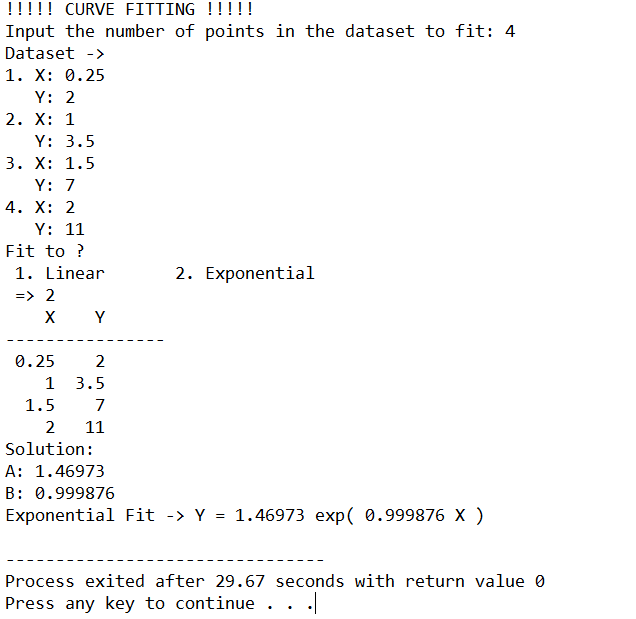
}

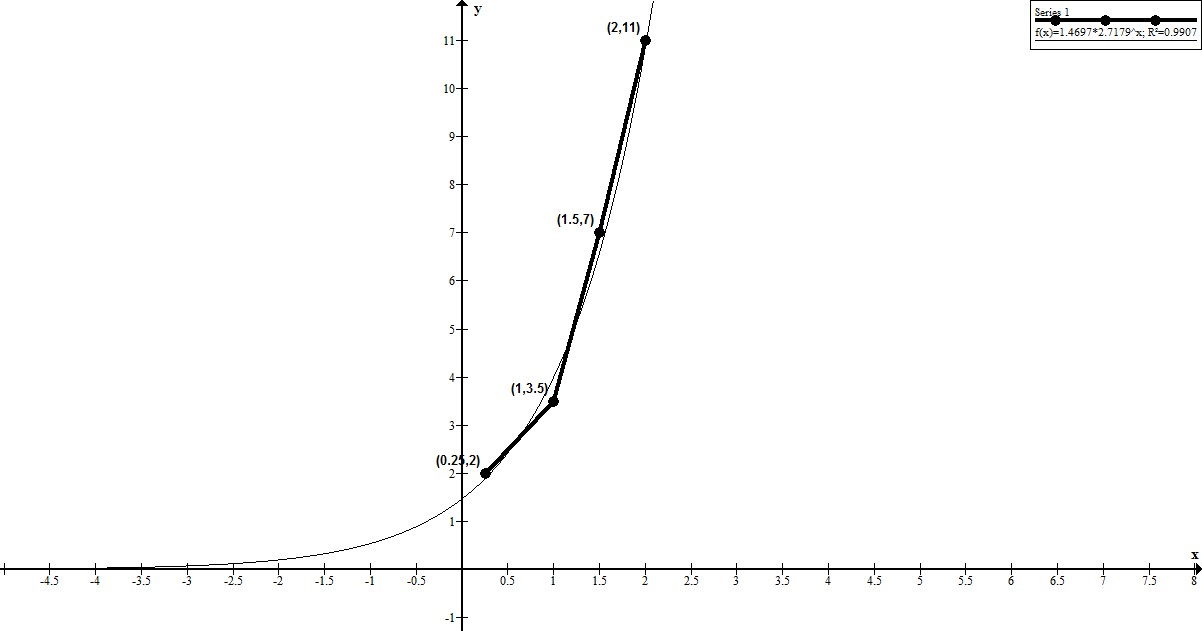
delete cf ;

return 0 ;

}

**OUTPUT:   
1. Linear Fit:   
  
**

**2. Exponential Fit:   
 ****

******

**DISCUSSION:**

**CONCLUSION:**