Jeremy Scheuerman

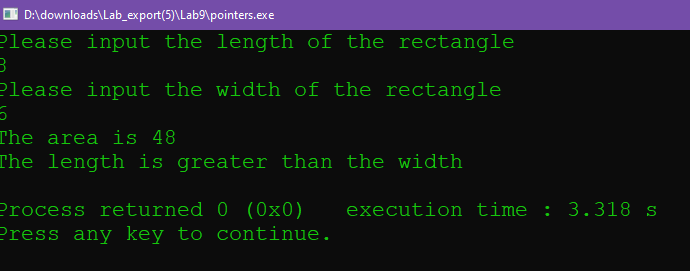
Dr. Wang

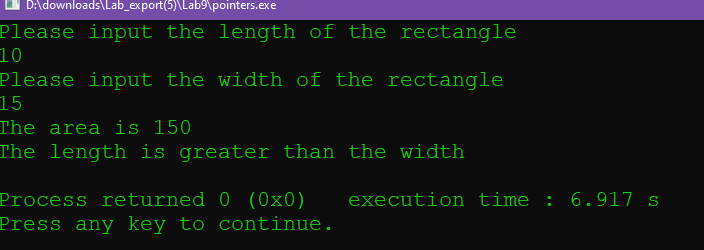
COSC 120

1 December 2020

Lab 9

9.1

1. 

2. 

IT says that the length is greater than the wdith

Source Code:

// This program demonstrates the use of pointer variables

// It finds the area of a rectangle given length and width

// It prints the length and width in ascending order

// PLACE YOUR NAME HERE

#include <iostream>

using namespace std;

int main()

{

int length; // holds length

int width; // holds width

int area; // holds area

int \*lengthPtr = nullptr; // int pointer which will be set to point to length

int \*widthPtr = nullptr; // int pointer which will be set to point to width

cout << "Please input the length of the rectangle" << endl;

cin >> length;

cout << "Please input the width of the rectangle" << endl;

cin >> width;

// Fill in code to make lengthPtr point to length (hold its address)

lengthPtr=&length;

// Fill in code to make widthPtr point to width (hold its address)

widthPtr=&width;

area =(\*lengthPtr)\*(\*widthPtr); // Fill in code to find the area by using only the pointer variables

cout << "The area is " << area << endl;

if (lengthPtr>widthPtr)

{

// Fill in the condition length > width by using only the pointer variables)

cout << "The length is greater than the width" << endl;

}

else if (widthPtr>lengthPtr)

{

// Fill in the condition of width > length by using only the pointer variables)

cout << "The width is greater than the length" << endl;

}

else

{

cout << "The width and length are the same" << endl;

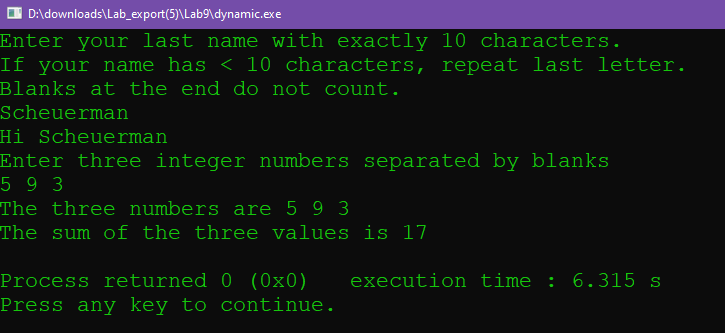
}

return 0;

}

9.2

1.



2. cin I think it would not work because of the way the pointer is set up

Cout I also think it would not work because of the way the pointer is setup

-I was incorrect and the program ran normally

Source Code:

// This program demonstrates the use of dynamic variables

// PLACE YOUR NAME HERE

#include <iostream>

using namespace std;

const int MAXNAME = 10;

int main()

{

int pos;

char \*name = nullptr;

int \*one = nullptr;

int \*two = nullptr;

int \*three = nullptr;

int result;

// Fill in code to allocate the integer variable one here

one =new int;

// Fill in code to allocate the integer variable two here

two=new int;

// Fill in code to allocate the integer variable three here

three=new int;

// Fill in code to allocate the character array pointed to by name

name =new char[10]

;

cout << "Enter your last name with exactly 10 characters." << endl;

cout << "If your name has < 10 characters, repeat last letter. " << endl

<< "Blanks at the end do not count." << endl;

for (pos = 0; pos < MAXNAME; pos++)

cin >>\*(name+pos); // Fill in code to read a character into the name array

// WITHOUT USING a bracketed subscript

cout << "Hi ";

for (pos = 0; pos < MAXNAME; pos++)

cout <<\*(name+pos); // Fill in code to a print a character from the name array

// WITHOUT USING a bracketed subscript

cout << endl << "Enter three integer numbers separated by blanks" << endl;

// Fill in code to input three numbers and store them in the

cin>>\*one;

cin>>\*two;

cin>>\*three;

// dynamic variables pointed to by pointers one, two, and three.

// You are working only with pointer variables

// echo print

cout << "The three numbers are " <<\*one <<" "<<\*two<<" "<<\*three<< endl;

// Fill in code to output those numbers

result =(\*one+\*two+\*three); // Fill in code to calculate the sum of the three numbers

cout << "The sum of the three values is " << result << endl;

delete one;

delete two;

delete three;

delete name;

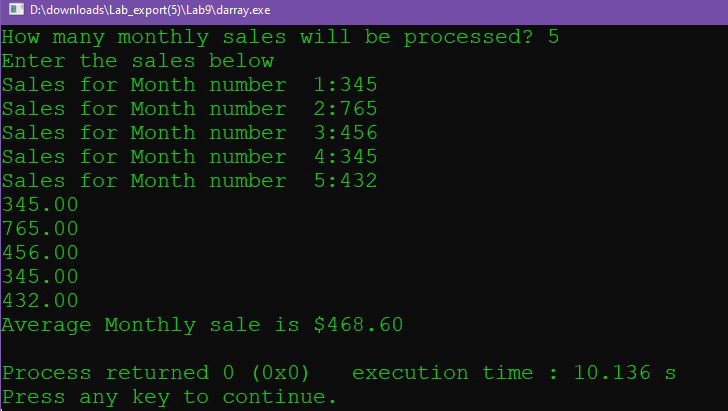
// Fill in code to deallocate one, two, three and name

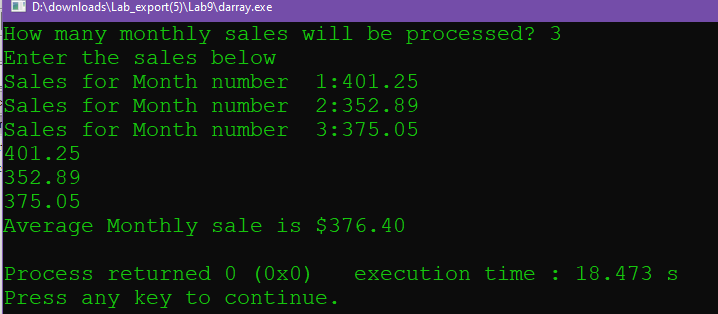
return 0;

}

9.3

1.



Source Code

// This program demonstrates the use of dynamic arrays

// PLACE YOUR NAME HERE

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

float \*monthSales = nullptr; // a pointer used to point to an array

// holding monthly sales

float total = 0; // total of all sales

float average; // average of monthly sales

int numOfSales; // number of sales to be processed

int count; // loop counter

cout << fixed << showpoint << setprecision(2);

cout << "How many monthly sales will be processed? ";

cin >> numOfSales;

// Fill in the code to allocate memory for the array pointed to by

// monthSales.

monthSales =new float[numOfSales];

if (monthSales==nullptr)

{

cout << "Error allocating memory!\n";

return 1;

}

// Fill in the condition to determine if memory has been

// allocated (or eliminate this if construct if your instructor

// tells you it is not needed for your compiler) )

cout << "Enter the sales below\n";

for (count = 0; count < numOfSales; count++)

{

cout << "Sales for Month number "

<< count+1// Fill in code to show the number of the month

<< ":";

cin>>(monthSales[count]);

// Fill in code to bring sales into an element of the array

}

for (count = 0; count < numOfSales; count++)

{

cout<<monthSales[count]<<endl;

total = total + monthSales[count];

}

average = total/numOfSales;// Fill in code to find the average

cout << "Average Monthly sale is $" <<fixed<<setprecision(2)<< average << endl;

// Fill in the code to deallocate memory assigned to the array.

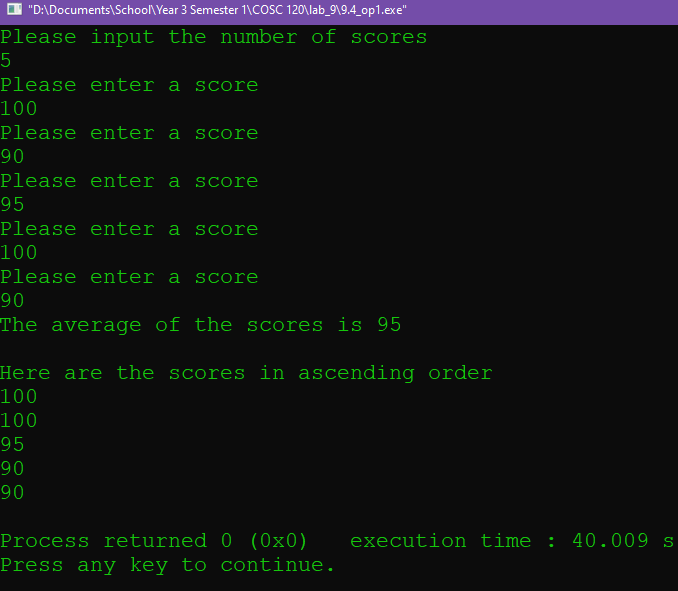
delete []monthSales;

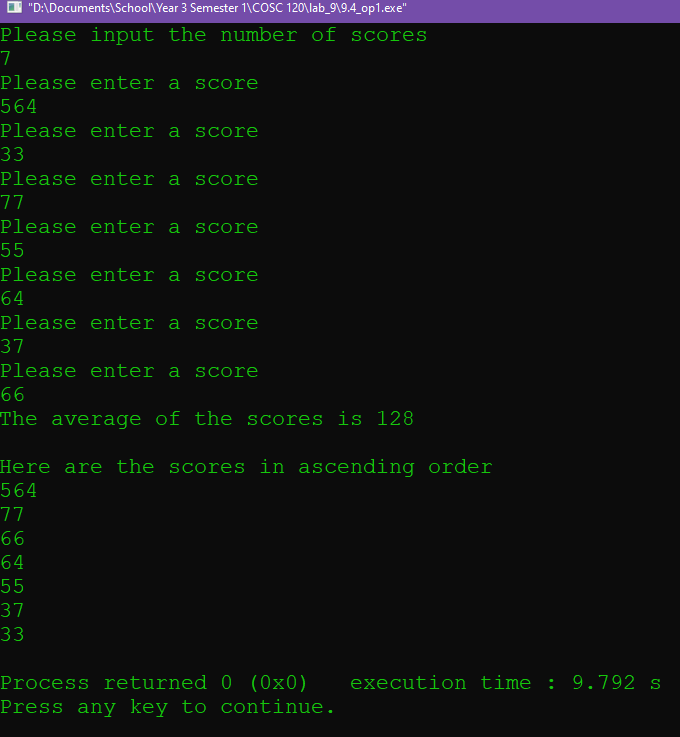
return 0;

}

9.4

Option 1





Source Code

#include <iostream>

using namespace std;

void sort(float\* score, int num\_scores)

{

//bubble sort to sort in ascending order

for (int i = 0; i < num\_scores-1; i++)

{

for (int j = 0; j < num\_scores-i-1; j++)

{

if (score[j] < score[j+1])

{

swap(score[j], score[j+1]);

//perform swap

}

}

}

cout<<"Here are the scores in ascending order"<<endl;

for (int i=0; i<num\_scores; i++)

{

//print swapped scores

cout<<score[i]<<endl;

}

}

int main()

{

float \*score = nullptr;

int num\_scores=0;

int total=0;

int avg=0;

// a pointer variable

cout<<"Please input the number of scores"<<endl;

cin>>num\_scores;

score =new float[num\_scores];

for (int i=0; i<num\_scores; i++)

{

cout<<"Please enter a score"<<endl;

cin>>score[i];

total+=score[i];

}

avg=total/num\_scores;

cout<<"The average of the scores is "<<avg<<endl<<endl;

sort(score,num\_scores);

// allocation of the array sort(score,scoreSize); // call to the function

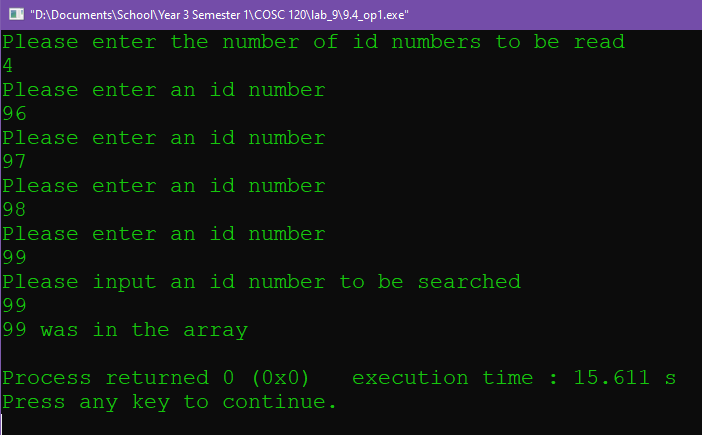
delete []score;

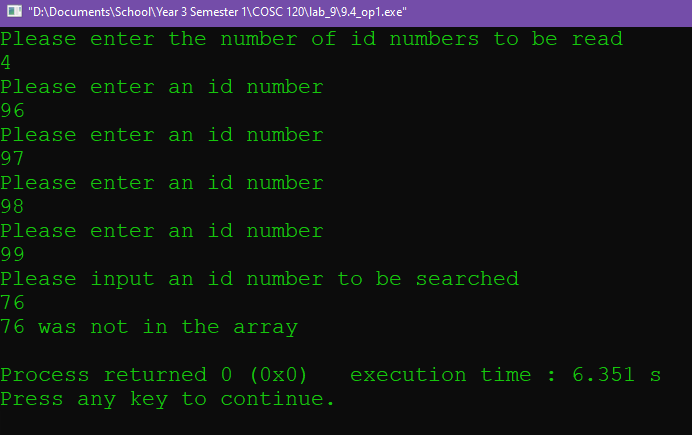
//deallocate

return 0;

}

Option 2





Source Code

#include <iostream>

using namespace std;

int main()

{

int\* ids=nullptr;

//define Ptr

int id\_amnt;

int id\_search;

bool in\_array;

//initialize

cout<<"Please enter the number of id numbers to be read"<<endl;

cin>>id\_amnt;

ids =new int [id\_amnt];

for (int i=0; i<id\_amnt; i++)

//populate array

{

cout<<"Please enter an id number"<<endl;

cin>>ids[i];

}

cout<<"Please input an id number to be searched"<<endl;

cin>>id\_search;

for(int i=0; i<<id\_amnt; i++)

//perform id search

{

if (ids[i]=id\_search)

{

in\_array=true;

break;

}

else

{

in\_array=false;

}

}

if (in\_array==true)

{

cout<<id\_search<<" was in the array"<<endl;

}

else

{

cout<<id\_search<<" was not in the array"<<endl;

}

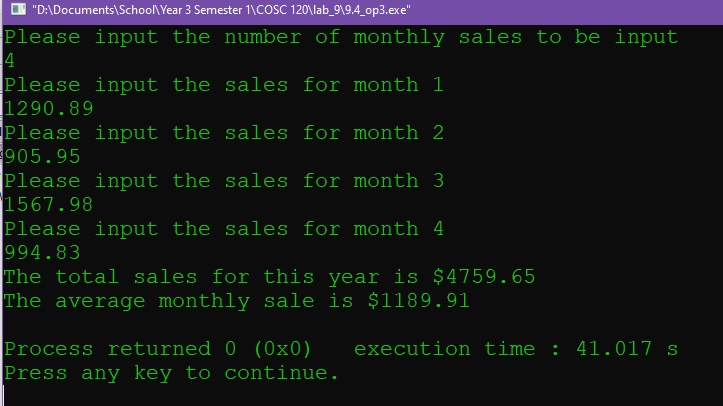
delete [] ids;

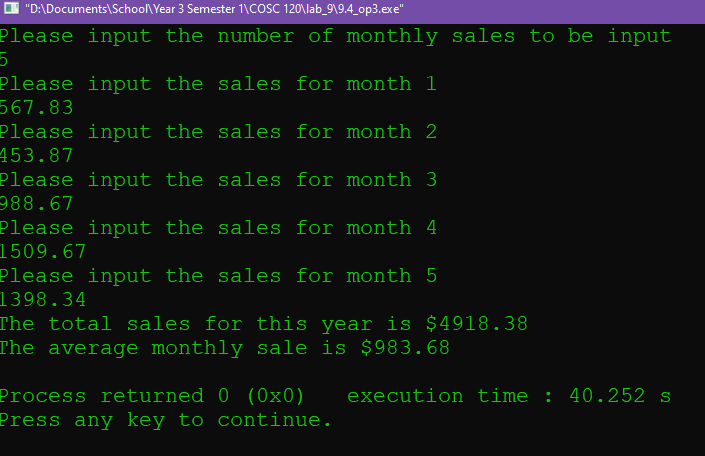
//deallocate

return 0;

}

Option 3





Source Code:

#include <iostream>

#include <iomanip>

using namespace std;

float get\_avg(float \*sales\_arr,int arr\_size);

float get\_total(float \*sales\_arr,int arr\_size);

float get\_avg(float \*sales\_arr,int arr\_size)

{

//function to calculate average with dynamic array

float total=0;

float avg=0;

for (int i=0; i<arr\_size; i++)

{

total+=sales\_arr[i];

}

avg=total/arr\_size;

//calculate and return

return avg;

}

float get\_total(float \*sales\_arr,int arr\_size)

{

//function to calculate average with dynamic array

float total=0;

for (int i=0; i<arr\_size; i++)

{

total+=sales\_arr[i];

}

//calculate and return

return total;

}

int main()

{

int months=0;

float \*Ptr=nullptr;

float total=0;

float avg=0;

//initialize

cout<<"Please input the number of monthly sales to be input "<<endl;

cin>>months;

Ptr=new float[months];

//allocate

for (int i=0; i<months; i++)

{

//populate array

cout<<"Please input the sales for month "<<i+1<<endl;

cin>>Ptr[i];

}

avg=get\_avg(Ptr,months);

total=get\_total(Ptr,months);

//get avg and total

cout<<"The total sales for this year is $"<<fixed<<setprecision(2)<<total<<endl;

cout<<"The average monthly sale is $"<<fixed<<setprecision(2)<<avg<<endl;

delete [] Ptr;

//deallocate memory

return 0;

}