CSC242 Introduction to Programming Concepts **Week Four Final Exam**

|  |  |
| --- | --- |
| **First Name** | Robert |
| **Last Name** | Dykes |
| **ID#** | 041502945 |
| **Email Address** |  |

Table of Contents

[Total Points = 100 1](#_Toc536753791)

[How to submit 2](#_Toc536753792)

[P1 - Array 2](#_Toc536753793)

[P2 - Series 3](#_Toc536753794)

[P3 – Inverted Pyramid of Odd Numbers 3](#_Toc536753795)

[P4 – Displaying Characters 4](#_Toc536753796)

# Total Points = 100

P1: 35 points

P2: 25 points

P3: 20 points

P4: 20 Points

# How to submit

After filling all the parts in this file, please follow the following steps.

1. Add your name and Student ID# to the first page.
2. Save the file in the original format (Docx or Doc)

(please do not convert to other file formats e.g. PDF, ZIP, RAR, …).

1. Rename the file as

*YOUR First Name-YOUR Last Name- ID-****Final****.docx*

**Example:**

John - Smith -234566435 *-* ***Final***.docx

1. Upload the file and submit it (only using Blackboard)

# P1 - Array

**Step 1 (10 points out of 35 points):** Write a function **isPrime()** that gets an integer number and returns **true** if the number is prime and returns **false** if the number is not prime.

**Hint:** A number is prime if it is divisible by only 1 and itself.

**Step 2 (15 points out of 35 points):** Write a function that gets an array and counts and returns the number of elements of the array that are prime (using the function in step 1).

**Step 3 (10 points out of 35 points):** In your main(), create an array of size 100 and initialize it with the first 100 odd numbers, starting 1 (that is 1, 3, 5, …). Use the function in step 2 to count the number of prime numbers in the array.

|  |
| --- |
| Your C++ code for this HW |
| #include <iostream>  using std::cin;  using std::cout;  using std::endl;  bool isPrime(int num);  int CountsPrime(int arr[]);  int main() {  int num, array[100];  int j=1;  for(int i = 0; i < 100; i++) {  array[i] = j;  j = j + 2;  }  cout << "\nYour array is :";  for (int i = 0;i < 100; i++)  cout << array[i] << " ";  cout << "Enter a whole positive number and press enter: ";  cin >> num;  if (isPrime(num)) {  cout << num <<" is a prime number" << endl;  } else {  cout << num <<" is not a prime number" << endl;  }  cout << "\nNumber of prime numbers in the array is: " << CountsPrime(array) << endl;  return 0;  }  bool isPrime(int num){  bool flag=true;  for(int i = 2; i <= num / 2; i++) {  if(num % i == 0) {  flag = false;  break;  }  }  return flag;  }  int CountsPrime(int array[])  {  int count=0;  for (int i = 0;i < 100 ;i++)  {  if (isPrime(array[i]))  count++;  }  return count;  } |

Then run the program and display the result.

|  |
| --- |
| Sample Run Result |
|  |

# P2 - Series

**Part 1 (5 points out of 25 points):** Write a **function** factorialFunc(n) that gets the positive integer **n** and calculates **n!**

n! = n\*(n-1)\* (n-2)\* (n-3) … \* 3 \* 2 \* 1;

**Part 2 (20 points out of 25 points): sin(x)** can be approximately calculated using the following formula, where n! is factorial(n) – for example 3!=3\*2\*1 = 6 (the function in previous problem).



The more terms we use in the series, the higher will be accuracy of the calculations. By using infinite terms in the series we will have the exact value.

Write a program that gets x and calculates **sin(x)** using 5, 10, 20 terms.

|  |
| --- |
| Your C++ code for this HW |
| #include <iostream>  #include <cmath>  using std::cin;  using std::cout;  using std::endl;  int factorial(int n) {  int result = 1;  int i = 0;  if( n <= 1)  return 1;  for(i = 2; i <= n; i++)  {  result = result \* i;  }  return result;  }  int main() {  int x;  cout << "Please enter \"x\" " << endl;  cin >> x;  cout << "factorial of x is: " << factorial(x) << endl;  cout << "The sin(x) is: " << sin(x) << endl;  } |

Then run the program and calculate and display the result for **x=pi/6**.

|  |
| --- |
| Sample Run Result |
|  |

# P3 – Inverted Pyramid of Odd Numbers

Write a program that prompts the user to enter an integer from 1 to 15 (that shows the number of lines) and displays an inverted pyramid of **odd** numbers starting with **1**, as shown in the following sample run:

Please enter the number of lines: 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 11 | 9 | 7 | 5 | 3 | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
|  | 11 | 9 | 7 | 5 | 3 | 1 | 3 | 5 | 7 | 9 | 11 |  |
|  |  | 9 | 7 | 5 | 3 | 1 | 3 | 5 | 7 | 9 |  |  |
|  |  |  | 7 | 5 | 3 | 1 | 3 | 5 | 7 |  |  |  |
|  |  |  |  | 5 | 3 | 1 | 3 | 5 |  |  |  |  |
|  |  |  |  |  | 3 | 1 | 3 |  |  |  |  |  |
|  |  |  |  |  |  | 1 |  |  |  |  |  |  |

Run the code and show the result for **input 9 (9 number of lines)**.

|  |
| --- |
| Your C++ code for this HW |
| #include <iostream>  using std::cin;  using std::cout;  using std:: endl;  int main() {  int rows,i,j,space;  cout << "Enter number of rows: ";  cin >> rows;  for (i = rows; i >= 1; --i) {  int count = i, counter = 1;  for(space = 0; space < rows -i; ++space)  cout << " ";  for (j = i; j <= 2 \* i - 1; ++j)  {  cout << 2 \* count - 1 << " ";  count--;  }  for ( j=0 ; j < i-1 ;++j)  {  cout<< 2 \* counter + 1 << " ";  counter++;  }  cout << endl;  }  return 0;  } |

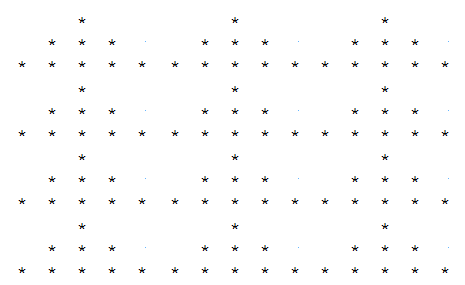
|  |
| --- |
| Sample Run Result |
|  |

# P4 – Displaying Characters

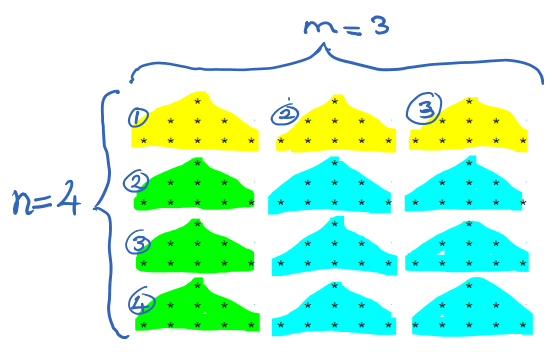
Consider the following block of character “\*” formed in the shape of a triangle.



Write a function that gets two input arguments ***n*** and ***m*** and then prints the above block n (row) \* m (column) times. For example for n = 4, and m = 3 the output should be as following.



The following figure shows it in more details.



|  |
| --- |
| Your C++ code for this problem |
| #include<iostream>  using std::cin;  using std::cout;  using std::endl;  void lotsOTriangles(int rows,int columns) {  for (int i = 0;i < rows;i++) {  for (int j = 0;j < columns;j++) {  for (int k = 0;k < 4;k++)  cout<<" ";  cout<<"\*";  for(int k = 0;k < 5;k++)  cout<<" ";  }  cout << endl;  for (int j = 0;j < columns ;j++){  for (int k = 0;k < 2;k++)  cout << " ";  cout << "\* \* \*";  for (int k = 0;k < 3;k++)  cout<<" ";  }  cout<< endl;  for(int j = 0;j < columns;j++)  {  cout<<"\* \* \* \* \*";  cout<<" ";  }  cout << endl;  }  }  int main() {  int rows;  int columns;  cout << "Please enter the number of rows and press enter: " << endl;  cin >> rows;  cout << "Please enter the number of columns and press enter: " << endl;  cin >> columns;  lotsOTriangles(rows,columns);  } |

Run the code for **n = 3 , m= 4** and insert the result in the following box.

|  |
| --- |
| The result of the query |
|  |

**The end**