

26.1 LAB: Track laps to miles



This section has been set as optional by your instructor.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

One lap around a standard high-school running track is exactly 0.25 miles. Define a function named LapsToMiles that takes a double as a parameter, representing the number of laps, and returns a double that represents the number of miles. Then, write a main program that takes a number of laps as an input, calls function LapsToMiles() to calculate the number of miles, and outputs the number of miles.

Output each floating-point value with two digits after the decimal point, which can be achieved by executing

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

 once before all other cout statements.

Ex: If the input is:

7.6

the output is:

1.90

Ex: If the input is:

2.2

the output is:

0.55

The program must define and call a function:

```
double LapsToMiles(double userLaps)
```

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

LAB
ACTIVITY

26.1.1: LAB: Track laps to miles



main.cpp

Load default template...

```
1 #include <iostream>
2 #include <iomanip>           // For setprecision
3 using namespace std;
```

```
4  
5 /* Define your function here */  
6  
7 int main() {  
8     /* Type your code here. Your code must call the function. */  
9  
10    return 0;  
11 }  
12
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.2 LAB: Step counter



This section has been set as optional by your instructor.

A pedometer treats walking 1 step as walking 2.5 feet. Define a function named FeetToSteps that takes a double as a parameter, representing the number of feet walked, and returns an integer that represents the number of steps walked. Then, write a main program that reads the number of feet walked as an input, calls function FeetToSteps() with the input as an argument, and outputs the number of steps.

Use floating-point arithmetic to perform the conversion.

Ex: If the input is:

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

150.5

the output is:

60

The program must define and call a function:

```
int FeetToSteps(double userFeet)
```

539740.3879454.qx3zqy7

LAB
ACTIVITY

26.2.1: LAB: Step counter

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here */
9
10    return 0;
11 }
12
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.3 LAB: A jiffy



This section has been set as optional by your instructor.

In computer animation, a "jiffy" is commonly defined as 1/100th of a second. Define a function named JiffiesToSeconds that takes a float as a parameter, representing the number of "jiffies", and returns a float that represents the number of seconds. Then, write a main program that reads the number of jiffies as an input, calls function JiffiesToSeconds() with the input as argument, and outputs the number of seconds.

Output each floating-point value with **three** digits after the decimal point, which can be achieved by executing

```
cout << fixed << setprecision(3);
```

once before all other cout statements.

Ex: If the input is:

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

the output is:

```
0.152
```

The program must define and call a function:

```
double JiffiesToSeconds(double userJiffies)
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.3.1: LAB: A jiffy

0 / 10



main.cpp

©zyBooks 01/31/24 18:14 1939727

[Load default template...](#)

MDCCOP2335Spring2024

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here. Your code must call the function. */
9
10    return 0;
11 }
```

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.4 LAB: Driving cost - functions

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024



This section has been set as optional by your instructor.

Write a function DrivingCost() with input parameters milesPerGallon, dollarsPerGallon, and milesDriven that returns the dollar cost to drive those miles. All items are of type double. The function called with arguments (20.0, 3.1599, 50.0) returns 7.89975.

Define that function in a program whose inputs are the car's miles per gallon and the price of gas in dollars per gallon (both doubles). Output the gas cost for 10 miles, 50 miles, and 400 miles, by calling your DrivingCost() function three times.

Output each floating-point value with two digits after the decimal point, which can be achieved by executing

`cout << fixed << setprecision(2);` once before all other cout statements.

The output ends with a newline.

Ex: If the input is:

20.0 3.1599

the output is:

1.58 7.90 63.20

Your program must define and call a function:

```
double DrivingCost(double milesPerGallon, double dollarsPerGallon, double milesDriven)
```

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024 0/10



LAB
ACTIVITY

26.4.1: LAB: Driving cost - functions

main.cpp

Load default template...

```
1 #include <iostream>
2 #include <iomanip> // For setprecision
3 using namespace std;
```

```
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here */
9
10    return 0;
11 }
12
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.5 LAB: Max magnitude



This section has been set as optional by your instructor.

Write a function MaxMagnitude() with three integer parameters that returns the largest magnitude value. Use the function in the main program that takes three integer inputs and outputs the largest magnitude value.

Ex: If the inputs are:

5 7 9

©zyBooks 01/31/24 18:14 1939727

function MaxMagnitude() returns and the main program outputs:

Rob Daglio
MDCCOP2335Spring2024

9

Ex: If the inputs are:

-17 -8 -2

function MaxMagnitude() returns and the main program outputs:

-17

Note: The function does not just return the largest value, which for -17 -8 -2 would be -2. Though not necessary, you may use the absolute-value built-in math function.

Your program must define and call a function:

```
int MaxMagnitude(int userVal1, int userVal2, int userVal3)
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.5.1: LAB: Max magnitude

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here */
9
10    return 0;
11 }
12
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.6 LAB: Count characters - functions



This section has been set as optional by your instructor.

Write a program whose input is a character and a string, and whose output indicates the number of times the character appears in the string. The output should include the input character and use the plural form, n's, if the number of times the character appears is not exactly 1.

Ex: If the input is:

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

n Monday

the output is:

1 n

Ex: If the input is:

```
z Today is Monday
```

the output is:

```
0 z's
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Ex: If the input is:

```
n It's a sunny day
```

the output is:

```
2 n's
```

Case matters. n is different than N.

Ex: If the input is:

```
n Nobody
```

the output is:

```
0 n's
```

The program must define and call the following function that takes the input string and character as parameters, and returns the number of times the input character appears in the input string.

```
int CalcNumCharacters(string userString, char userChar)
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.6.1: LAB: Count characters - functions

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 using namespace std;
3
4 /* Define your function here */
5
6 int main() {
7     /* Type your code here. Your code must call the function. */
8
9     return 0;
10}
11
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.7 LAB: Remove all non-alphabetic characters - functions



This section has been set as optional by your instructor.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Write a program that removes all non-alphabetic characters from the given input.

Ex: If the input is:

-Hello, 1 world\$!

the output is:

Helloworld

Your program must define and call the following function that takes a string as a parameter and returns the string without any non-alphabetic characters.

string RemoveNonAlpha(string userString)

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

**LAB
ACTIVITY**

26.7.1: LAB: Remove all non-alphabetic characters - functions

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 using namespace std;
3
4 /* Define your function here */
5
6 int main() {
7
8     /* Type your code here. Your code must call the function. */
9
10    return 0;
11 }
12
```

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Run program

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.8 LAB: Convert to binary - functions



This section has been set as optional by your instructor.

Write a program that takes in a positive integer as input, and outputs a string of 1's and 0's representing the integer in binary. For an integer x , the algorithm is:

```
As long as x is greater than 0
    Output x % 2 (remainder is either 0 or 1)
    x = x / 2
```

Note: The above algorithm outputs the 0's and 1's in reverse order. You will need to write a second function to reverse the string.

Ex: If the input is:

6

the output is:

110

The program must define and call the following two functions. Define a function named `IntToReverseBinary()` that takes an int as a parameter and returns a string of 1's and 0's representing the integer in binary (in reverse). Define a function named `StringReverse()` that takes a string as a parameter and returns a string representing the input string in reverse.

```
string IntToReverseBinary(int integerValue)
string StringReverse(string userString)
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.8.1: LAB: Convert to binary - functions

0 / 10



```
1 #include <iostream>
2 using namespace std;
3
4 /* Define your functions here */
5
6 int main() {
7     /* Type your code here.
8      Your code must call IntToReverseBinary() to get
9      the binary string of an integer in a reverse order.
10     Then call StringReverse() to reverse the string
11     returned from IntToReverseBinary().*/
12
13
14     return 0;
15 }
```

@zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work

[What is this?](#)

@zyBooks 01/31/24 18:14 1939727

History of your effort will appear here once you begin working
on this zyLab.

Rob Daglio
MDCCOP2335Spring2024

26.9 LAB: Max and min numbers



This section has been set as optional by your instructor.

Write a program whose inputs are four integers, and whose outputs are the maximum and the minimum of the four values.

Ex: If the input is:

12 18 4 9

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

the output is:

Maximum is 18
Minimum is 4

The program must define and call the following two functions. Define a function named MaxNumber that takes four integer parameters and returns an integer representing the maximum of the four integers. Define a function named MinNumber that takes four integer parameters and returns an integer representing the minimum of the four integers.

```
int MaxNumber(int num1, int num2, int num3, int num4)  
int MinNumber(int num1, int num2, int num3, int num4)
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.9.1: LAB: Max and min numbers

0 / 10



main.cpp

Load default template...

```
1 #include <iostream>  
2 using namespace std;  
3  
4 /* Define your functions here */  
5  
6 int main() {  
7     /* Type your code here. Your code must call the functions. */  
8  
9     return 0;  
10 }  
11
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024 //

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.10 LAB: Acronyms



This section has been set as optional by your instructor.

An acronym is a word formed from the initial letters of words in a set phrase. Define a function named CreateAcronym that takes a string parameter and returns the acronym of the string parameter. Append a period (.) after each letter in the acronym. If a word begins with a lower case letter, don't include that letter in the acronym. Then write a main program that reads a phrase from input, calls CreateAcronym() with the input phrase as argument, and outputs the returned acronym. Assume the input has at least one upper case letter.

Ex: If the input is:

Institute of Electrical and Electronics Engineers

the output should be:

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

I.E.E.E.

Ex: If the input is:

Association for computing MACHINERY

the output should be:

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

A.M.

Although the letters ACHINERY in MACHINERY are upper case, those letters are omitted for being a part of the word MACHINERY.

The program must define and call a function:

string CreateAcronym(string userPhrase)

Hint: Use isupper() to check if a letter is upper case.

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.10.1: LAB: Acronyms

0 / 10

**main.cpp****Load default template...**

```
1 #include <iostream>
2 #include <string>
3 #include <cctype>
4 using namespace std;
5
6 /* Define your function here */
7
8 int main() {
9     /* Type your code here */
10
11     return 0;
12 }
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.11 LAB: Leap year - functions



This section has been set as optional by your instructor.

A common year in the modern Gregorian Calendar consists of 365 days. In reality, Earth takes longer to rotate around the sun. To account for the difference in time, every 4 years, a leap year takes place. A leap year is when a year has 366 days: An extra day, February 29th. The requirements for a given year to be a leap year are:

- 1) The year must be divisible by 4
- 2) If the year is a century year (1700, 1800, etc.), the year must be evenly divisible by 400; therefore, both 1700 and 1800 are not leap years

Some example leap years are 1600, 1712, and 2016.

Write a program that takes in a year and determines the number of days in February for that year.

Ex: If the input is:

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

1712

the output is:

1712 has 29 days in February.

Ex: If the input is:

1913

the output is:

1913 has 28 days in February.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

Your program must define and call the following function. The function should return the number of days in February for the input year.

```
int DaysInFeb(int userYear)
```

539740.3879454.qx3zqy7

LAB
ACTIVITY

26.11.1: LAB: Leap year - functions

0 / 10



main.cpp

Load default template...

```
1 #include <iostream>
2 using namespace std;
3
4 /* Define your function here */
5
6 int main() {
7     /* Type your code here. Your code must call the function. */
8
9     return 0;
10 }
11
```

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.12 LAB: Exact change - functions



This section has been set as optional by your instructor.

Define a function called ExactChange that takes the total change amount in cents and an integer vector as parameters. Function ExactChange() calculates the change using the fewest coins and stores the number of each coin type used into the vector parameter. Index 0-3 of the vector parameter should contain the number of pennies, nickels, dimes, and quarters respectively. Then write a main program that reads the total change amount as an integer input, calls ExactChange(), and outputs the change, one coin type per line. Use singular and plural coin names as appropriate, like 1 penny vs. 2 pennies. Output "no change" if the input is 0 or less.

Ex: If the input is:

0

(or less), the output is:

no change

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Ex: If the input is:

45

the output is:

```
2 dimes  
1 quarter
```

Your program must define and call the following function. Positions 0-3 of coinVals should contain the number of pennies, nickels, dimes, and quarters, respectively.

```
void ExactChange(int userTotal, vector<int>& coinVals)
```

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

0 / 10



LAB
ACTIVITY

26.12.1: LAB: Exact change - functions

main.cpp

Load default template...

```
1 #include <iostream>  
2 #include <vector>  
3 using namespace std;  
4  
5 /* Define your function here */  
6  
7 int main() {  
8     int inputVal;  
9     vector<int> changeAmount(4);  
10  
11    /* Type your code here. Your code must call the function. */  
12  
13    return 0;  
14 }  
15
```

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Run program

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio
MDCCOP2335Spring2024

26.13 LAB: Multiples of ten in a vector



This section has been set as optional by your instructor.

Write a program that reads a list of integers, and outputs whether the list contains all multiples of 10, no multiples of 10, or mixed values. Define a function named IsVectorMult10 that takes a vector as a parameter, representing the list, and returns a boolean that represents whether the list contains all multiples of ten. Define a function named IsVectorNoMult10 that takes a vector as a parameter, representing the list, and returns a boolean that represents whether the list contains no multiples of ten.

Then, write a main program that takes an integer, representing the size of the list, followed by the list values. The first integer is not in the list.

Ex: If the input is:

```
5 20 40 60 80 100
```

the output is:

```
all multiples of 10
```

Ex: If the input is:

```
5 11 -32 53 -74 95
```

the output is:

```
no multiples of 10
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio
MDCCOP2335Spring2024

Ex: If the input is:

```
5 10 25 30 40 55
```

the output is:

mixed values

The program must define and call the following two functions. IsVectorMult10 returns true if all integers in the vector are multiples of 10 and false otherwise. IsVectorNoMult10 returns true if no integers in the vector are multiples of 10 and false otherwise.

```
bool IsVectorMult10(vector<int> myVec)
bool IsVectorNoMult10(vector<int> myVec)
```

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

LAB
ACTIVITY

26.13.1: LAB: Multiples of ten in a vector

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 /* Define your functions here */
6
7 int main() {
8     /* Type your code here. */
9
10    return 0;
11 }
```

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

[Run program](#)

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.14 LAB: Contact list



This section has been set as optional by your instructor.

A contact list is a place where you can store a specific contact with other associated information such as a phone number, email address, birthday, etc. Write a program that first takes as input an integer N that represents the number of word pairs in the list to follow. Word pairs consist of a name and a phone number (both strings), separated by a comma. That list is followed by a name, and your program should output the phone number associated with that name. Output "None" if name is not found.

Ex: If the input is:

```
3 Joe,123-5432 Linda,983-4123 Frank,867-5309 Frank
```

the output is:

```
867-5309
```

Your program must define and call the following GetPhoneNumber() function. The return value of GetPhoneNumber() is the phone number associated with the specific contact name.

GetPhoneNumber() returns "None" if name is not found, and the main program outputs the returned value accordingly.

```
string GetPhoneNumber(vector<string> nameVec, vector<string>  
phoneNumberVec, string contactName)
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Hint: Use two vectors: One for the string names, and the other for the string phone numbers

539740.3879454.qx3zqy7

LAB
ACTIVITY

26.14.1: LAB: Contact list

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here */
9
10    return 0;
11 }
12
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working
on this zyLab.

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

26.15 LAB: Output values below an amount - functions



This section has been set as optional by your instructor.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Write a program that first gets a list of integers from input. The input begins with an integer indicating the number of integers that follow. Then, get the last value from the input, and output all integers less than or equal to that value.

Ex: If the input is:

```
5
50 60 140 200 75
100
```

the output is:

```
50 60 75
```

The 5 indicates that there are five integers in the list, namely 50, 60, 140, 200, and 75. The 100 indicates that the program should output all integers less than or equal to 100, so the program outputs 50, 60, and 75. For coding simplicity, follow every output value by a space, including the last one.

Such functionality is common on sites like Amazon, where a user can filter results. Utilizing functions helps to make main() very clean and intuitive.

The program must define the following two functions:

`void GetUserValues(vector<int>& userValues, int numValues)` - read `numValues` integers and store the integers in `userValues`.

`void IntsLessThanOrEqualToThreshold(vector<int> userValues, int upperThreshold, vector<int>& resValues)` - store in `resValues` values of `userValues` that are less than or equal to `upperThreshold`.

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024 0/10

LAB
ACTIVITY

26.15.1: LAB: Output values below an amount - functions

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
```

main.cpp

[Load default template...](#)

```
3 using namespace std;
4
5 /* Define your functions here */
6
7 int main() {
8     vector<int> userValues;
9     vector<int> resValues;
10    int upperThreshold;
11    int numValues;
12    unsigned int i;
13
14    cin >> numValues;
15    GetUserValues(userValues, numValues);
16}
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.16 LAB: Subtracting list elements from max - functions



This section has been set as optional by your instructor.

When analyzing data sets, such as data for human heights or for human weights, a common step is to adjust the data. This can be done by normalizing to values between 0 and 1, or throwing away outliers. For this program, adjust the values by subtracting each value from the maximum. The input begins with an integer indicating the number of integers that follow. Assume that the list will always contain between 1 and 20 integers.

Ex: If the input is:

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

5 30 50 10 70 65

the output is:

40 20 60 0 5

For coding simplicity, follow every output value by a space, even the last one.

Your program must define and use the following function:

```
int GetMaxInt(vector<int> listInts)
```

539740.3879454.qx3zqy7

LAB ACTIVITY

26.16.1: LAB: Subtracting list elements from max - functions

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 /* Define your function here */
6
7 int main() {
8     /* Type your code here */
9
10    return 0;
11 }
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp

(Your program)

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.17 LAB: Replacement words



This section has been set as optional by your instructor.

Write a program that finds word differences between two sentences. The input begins with an integer indicating the number of words in each sentence. The next input line is the first sentence and the following input line is the second sentence. Assume that the two sentences have the same number of words and no more than 20 words each.

The program displays word pairs that differ between the two sentences. One pair is displayed per line.

Ex: If the input is:

6

Smaller cars get better gas mileage
Tiny cars get great fuel economy

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

then the output is:

Smaller Tiny
better great

```
gas fuel  
mileage economy
```

Add a function named ReadSentences to read the input sentences into two string vectors.

```
void ReadSentences(vector<string>& sentence1Words, vector<string>&  
sentence2Words, int wordCount)
```

main() already contains code to read the word count and call ReadSentences(). Complete main() to display differing word pairs.

539740.3879454.qx3zqy7

Rob Daglio
MDCCOP2335Spring2024

**LAB
ACTIVITY**

26.17.1: LAB: Replacement words

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>  
2 #include <vector>  
3 #include <string>  
4 using namespace std;  
5  
6 /* Define your function here */  
7  
8 int main() {  
9     vector<string> firstSentence(20);  
10    vector<string> secondSentence(20);  
11    int wordCount;  
12    int i;  
13  
14    cin >> wordCount;  
15    // Call ReadSentences() to populate vectors
```

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

[Run program](#)

Input (from above)



main.cpp
(Your program)



Output

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.18 LAB: Toll calculations



This section has been set as optional by your instructor.

Toll roads have different fees at different times of the day and on weekends. Write a function CalcToll() that has three arguments: the current hour of time (int), whether the time is morning (bool), and whether the day is a weekend (bool). The function returns the correct toll fee (double), based on the chart below.

Weekday Tolls

- Before 7:00 am (\$6.15)
- 7:00 am to 9:59 am (\$7.95)
- 10:00 am to 2:59 pm (\$6.90)
- 3:00 pm to 7:59 pm (\$8.95)
- Starting 8:00 pm (\$6.40)

Weekend Tolls

- Before 7:00 am (\$6.05)
- 7:00 am to 7:59 pm (\$7.15)
- Starting 8:00 pm (\$6.10)

Ex: The function calls below, with the given arguments, will return the following toll fees:

CalcToll(8, true, false) returns 7.95

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

CalcToll(1, false, false) returns 6.90

CalcToll(3, false, true) returns 7.15

CalcToll(5, true, true) returns 6.05

539740.3879454.qx3zqy7



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 double CalcToll(int hour, bool isMorning, bool isWeekend) {
6     /* Type your code here */
7 }
8
9 int main() {
10    cout << CalcToll(8, true, false) << endl; // 7.95
11    cout << CalcToll(1, false, false) << endl; // 6.9
12    cout << CalcToll(3, false, true) << endl; // 7.15
13    cout << CalcToll(5, true, true) << endl; // 6.05
14
15    return 0;
16 }
```

[Develop mode](#)[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

[Run program](#)

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work

[What is this?](#)

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

History of your effort will appear here once you begin working on this zyLab.

26.19 LAB: Calculate average



This section has been set as optional by your instructor.

Complete the CalcAverage() function that has a double vector parameter and returns the average value of the elements in the vector as an integer.

© Rob Daglio 01/31/24 10:14 1030727
Rob Daglio
MDCCOP2335Spring2024

Ex: If the input vector is:

```
1.1 2.1 3.1 4.1 5.1
```

then the returned average will be:

```
3
```

539740.3879454.qx3zqy7

LAB ACTIVITY

26.19.1: LAB: Calculate average

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 // CalcAverage() takes in a double vector and returns the average value of each element
6 int CalcAverage(vector<double> nums) {
7
8     /* Type your code here. */
9
10 }
11
12 int main() {
13     vector<double> nums = {1.1, 2.1, 3.1, 4.1, 5.1};
14     cout << CalcAverage(nums) << endl; // CalcAverage() should return 3.39727
15 }
```

Rob Daglio

MDCCOP2335Spring2024

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.20 LAB: Count evens



This section has been set as optional by your instructor.

Write a function CountEvens() that has five integer parameters, and returns the count of parameters where the value is an even number (i.e. evenly divisible by 2).

Ex: If the five parameters are:

1 22 11 40 37

then the returned count will be:

2

©zyBooks 01/31/24 18:14 1939727

Hint: Use the modulo operator % to determine if each number is even or odd.

Rob Daglio

MDCCOP2335Spring2024

Your program must define the function:

```
int CountEvens(int num1, int num2, int num3, int num4, int num5)
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.20.1: LAB: Count evens

0 / 10



main.cpp

1 Loading latest submission...

©zyBooks 01/31/24 18:14 1939727

Rob Daglio
MDCCOP2335Spring2024**Develop mode****Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work

[What is this?](#)

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024



Retrieving signature

26.21 LAB: Reverse vector



This section has been set as optional by your instructor.

Complete Reverse() function that returns a new integer vector containing all contents in the vector parameter reversed.

Ex: If the elements of the input vector are:

```
{2, 4, 6}
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

the returned array will be:

```
{6, 4, 2}
```

539740.3879454.qx3zqy7

LAB
ACTIVITY

26.21.1: LAB: Reverse vector

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 // This method reverses contents of parameter vect.
6 vector<int> Reverse(vector<int> vect) {
7
8     // Type your code here.
9
10 }
11
12 int main() {
13     size_t i = 0;
14     vector<int> intVect(3);
15     intVect.at(0) = 2;
16 }
```

[Develop mode](#)

[Submit mode](#)

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

@zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.22 LAB: Check if vector is sorted



This section has been set as optional by your instructor.

Write the InOrder() function, which receives a vector of integers as a parameter, and returns true if the numbers are sorted in descending order (in order from high to low) or false otherwise. The program outputs "In descending order" if the vector is sorted, or "Not in order" if the vector is not sorted.

Ex: If the vector passed to the InOrder() function is [5, 6, 7, 8, 3], then the function returns false and the program outputs:

Not in order

Ex: If the vector passed to the InOrder() function is [10, 8, 7, 6, 5], then the function returns true and the program outputs:

In descending order

539740.3879454.qx3zqy7

@zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

**LAB
ACTIVITY**

26.22.1: LAB: Check if vector is sorted

0/10

main.cpp[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
```

```
3 using namespace std;
4
5
6 bool InOrder(vector<int> nums) {
7     /* Type your code here */
8 }
9
10 int main() {
11
12     vector<int> nums1 = {5, 6, 7, 8, 3};
13
14     if (InOrder(nums1)) {
15         cout << "In descending order" << endl;
16 }
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

26.23 LAB: Remove all odd numbers from a vector



This section has been set as optional by your instructor.

Write the RemoveOdds() function, which receives a vector of integers as a parameter and returns a new vector of integers containing only the even numbers from the original vector. The main program outputs values of the returned vector.

Hint: If the original vector has odd numbers, then the new vector will be smaller in length than the original vector and should have no blank element.

Ex: If the vector passed to the RemoveOdds() function is [1, 2, 3, 4, 5, 6, 7, 8, 9], then the function returns and the program output is:

```
[2, 4, 6, 8]
```

Ex: If the vector passed to the RemoveOdds() function is [2, 8, 6], then the function returns and the program output is:

```
[2, 8, 6]
```

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.23.1: LAB: Remove all odd numbers from a vector

0 / 10

main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 vector<int> RemoveOdds(vector<int> nums) {
6     /* Type your code here */
7
8 }
9
10 int main() {
11     vector<int> input = {1,2,3,4,5,6,7,8,9};
12
13     vector<int> result = RemoveOdds(input); // Should return [2, 4, 6, 8]
14
15     cout << "[";
16 }
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.24 LAB: Fun with characters



This section has been set as optional by your instructor.

Complete the CheckCharacter() function which has 2 parameters: A string and a specified index (int). The function checks the character at the specified index of the string parameter. The function then returns a string based on the type of character at that location indicating if the character is a letter, digit, whitespace, or unknown character.

Hint: Use the functions isalpha(), isspace() and isdigit() in your solution.

Ex: The function calls below with the given arguments will return the following strings:

CheckCharacter("happy birthday", 2) returns "Letter: 'p'"

CheckCharacter("happy birthday", 5) returns "Whitespace: ' '"

CheckCharacter("happy birthday 2 you", 15) returns "Digit: '2'"

CheckCharacter("happy birthday!", 14) returns "Unknown: '!'"

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

**LAB
ACTIVITY**

26.24.1: LAB: Fun with characters

0 / 10

**main.cpp**[Load default template...](#)

```
1 #include <iostream>
2 using namespace std;
3
4 string CheckCharacter(string word, int index) {
5     /* Type your code here */
6 }
7
8 int main() {
9     cout << CheckCharacter("happy birthday", 2) << endl;    Rob Daglio
10    cout << CheckCharacter("happy birthday", 5) << endl;   MDCCOP2335Spring2024
11    cout << CheckCharacter("happy birthday 2 you", 15) << endl;
12    cout << CheckCharacter("happy birthday!", 14) << endl;
13
14    return 0;
15 }
```

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

→ Output

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

26.25 LAB: Write SumAndCast() function to cast double to int



This section has been set as optional by your instructor.

Complete the SumAndCast() function that casts the parameters from doubles to integers and returns the resulting sum.

Note that the main() function prints out the returned value of the SumAndCast() function.

Ex: If the double values are 14.2 and 19.9, then the output is:

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

33

Ex: If the double values are 2.5 and 3.1, then the output is:

5

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.25.1: LAB: Write SumAndCast() function to cast double to int

0 / 10



main.cpp

[Load default template...](#)

```
1 #include <iostream>
2 using namespace std;
3
4 int SumAndCast(double d1, double d2) {
5
6     /* Type your code here */
7
8 }
9
10 int main() {
11     cout << SumAndCast(14.2, 19.9) << endl;
12     cout << SumAndCast(2.5, 3.1) << endl;
13
14     return 0;
15 }
```

[Develop mode](#)

[Submit mode](#)

©zyBooks 01/31/24 18:14 1939727

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

main.cpp
(Your program)

Output

Program output displayed here

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.26 LAB*: Program: Poker dice game



This section has been set as optional by your instructor.

Program Specifications Write a program to calculate the score from a throw of five dice. Scores are assigned to different categories for singles, three of a kind, four of a kind, five of a kind, full house, and straight. Follow each step to gradually complete all functions.

Note: This program is designed for *incremental development*. Complete each step and submit for grading before starting the next step. Only a portion of tests pass after each step but confirm progress.

Step 0. Review the provided main() function. Five integer values are input and inserted into a vector. The vector is sorted and passed to FindHighScore() to determine the highest scoring category. Make no changes to main(). Stubs are provided for all remaining functions.

Step 1 (3 pts). Complete the CheckSingles() function. Return the sum of all values that match parameter goal. Update the FindHighScore() function using a loop to call CheckSingles() six times with parameters being 1 - 6. Return the highest score from all function calls. Submit for grading to confirm two tests pass.

Ex: If input is:

2 4 1 5 4

the output is:

High score: 8

Step 2 (3 pts). Complete the CheckThreeOfKind(), CheckFourOfKind(), and CheckFiveOfKind() functions. Hint: Since the values are in ascending order, same values are stored in consecutive index locations. Return 30 from CheckThreeOfKind() if the dice contain at least three of the same values. Ex: (2, 3, 3, 3, 6). Return 40 from CheckFourOfKind() if the dice contain at least four of the same values. Ex: (4, 4, 4, 4, 5). Return 50 from CheckFiveOfKind() if the dice contain five identical values. Ex: (5, 5, 5, 5, 5). Update the FindHighScore() function to call the three functions and return the highest score from all function calls. Submit for grading to confirm five tests pass.

Ex: If input is:

2 4 4 5 4

the output is:

High score: 30

Step 3 (2 pts). Complete the CheckFullHouse() function to return 35 if the dice contain a full house (a pair and three of a kind). Ex: (1, 1, 3, 3, 3). Note: Five of a kind also satisfies the definition of a full house since (4, 4, 4, 4, 4) includes a pair of 4s and three 4s. Update the FindHighScore() function to call CheckFullHouse() and return the highest score from all function calls. Submit for grading to confirm seven tests pass.

Step 4 (2 pts). Complete the CheckStraight() function to return 45 if the dice contain a straight of (1, 2, 3, 4, 5) or (2, 3, 4, 5, 6). Update the FindHighScore() function to call CheckStraight() and return the highest score from all function calls. Submit for grading to confirm all tests pass.

539740.3879454.qx3zqy7

LAB
ACTIVITY

26.26.1: LAB*: Program: Poker dice game

0 / 10



main.cpp

Load default template...

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5
6 // Add all occurrences of goal value
7 int CheckSingles(vector<int>& diceValues, int goal) {
8     /* Complete the function and update the return statement */
9     return -1;
10 }
11
```

©zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

```
12 // Check for three of a kind (score = 30)
13 int CheckThreeOfKind(vector<int>& diceValues) {
14     /* Complete the function and update the return statement */
15     return -1;
16 }
```

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

@zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**
(Your program)

Output

Program output displayed here

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

26.27 LAB*: Program: Income tax form - functions



This section has been set as optional by your instructor.

@zyBooks 01/31/24 18:14 1939727
Rob Daglio
MDCCOP2335Spring2024

Program Specifications Write a program to calculate U.S. income tax owed given wages, taxable interest, unemployment compensation, status (dependent, single, or married), and taxes withheld. Dollar amounts are displayed as integers without comma separators. For example, `cout << "Deduction: $" << deduction << endl;`

Note: this program is designed for *incremental development*. Complete each step and submit for

grading before starting the next step. Only a portion of tests pass after each step but confirm progress.

Step 1. Within main() input wages, taxable interest, unemployment compensation, status (0=dependent, 1=single, and 2=married), and taxes withheld as integers.

Step 2 (2 pts). Complete the CalcAGI() function. Calculate the adjusted gross income (AGI) that is the sum of wages, interest, and unemployment. Convert any negative values to positive before summing to correct potential input errors. Return the AGI. Note the provided code in main() calls CalcAGI() and outputs the returned value. Submit for grading to confirm two tests pass.

Ex: If the input is:

```
20000 23 500 1 400
```

The output is:

```
AGI: $20523
```

Step 3 (2 pts). Complete the GetDeduction() function. Return the deduction amount based on status: (0) dependent = 6000, (1) single = 12000, or (2) married=24000. Return 6000 if the status is anything but 0, 1, or 2. Within main() call GetDeduction() and output the returned value. Submit for grading to confirm four tests pass.

Ex: If the input is:

```
20000 23 500 1 400
```

The additional output is:

```
AGI: $20523  
Deduction: $12000
```

Step 4 (2 pts). Complete the CalcTaxable() function. Calculate taxable amount (AGI - deduction). Set taxable to zero if calculation results in negative value. Return taxable value. Within main() call CalcTaxable() and output the returned value. Submit for grading to confirm six tests pass.

Ex: If the input is:

```
20000 23 500 1 400
```

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

The additional output is:

```
AGI: $20523  
Deduction: $12000  
Taxable income: $8523
```

Step 5 (2 pts). Complete the CalcTax() function. Calculate tax amount based on status and taxable income (see tables below). Tax amount should be stored initially as a double, rounded to the nearest whole number using round(), and converted to an integer before returning. Within main() call CalcTax() and output the returned value. Submit for grading to confirm eight tests pass.

Ex: If the input is:

50000 0 0 2 5000

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

The additional output is:

AGI: \$50000
 Deduction: \$24000
 Taxable income: \$26000
 Federal tax: \$2720

| Income | Tax for Dependent or Single Filers |
|-------------------|--|
| \$0 - \$10000 | 10% of the income |
| \$10001 - \$40000 | \$1000 + 12% of the amount over \$10000 |
| \$40001 - \$85000 | \$4600 + 22% of the amount over \$40000 |
| over \$85000 | \$14500 + 24% of the amount over \$85000 |

| Income | Tax for Married Filers |
|-------------------|---|
| \$0 - \$20000 | 10% of the income |
| \$20001 - \$80000 | \$2000 + 12% of the amount over \$20000 |
| over \$80000 | \$9200 + 22% of the amount over \$80000 |

Step 6 (2 pts). Complete the CalcTaxDue() function. Set withheld parameter to zero if negative to correct potential input error. Calculate and return amount of tax due (tax - withheld). Within main() call CalcTaxDue() and output returned value. Submit for grading to confirm all tests pass.

Ex: If the input is:

80000 0 500 2 12000

©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

The additional output is:

AGI: \$80500
 Deduction: \$24000

Taxable income: \$56500

Federal tax: \$6380

Tax due: \$-5620

539740.3879454.qx3zqy7

**LAB
ACTIVITY**

26.27.1: LAB*: Program: Income tax form - functions

0 / 10



©zyBooks 01/31/24 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

[Load default template...](#)

main.cpp

```
1 #include <iostream>
2 #include <iomanip>
3 #include <cmath>
4 using namespace std;
5
6 // Calculate AGI and repair any negative values
7 int CalcAGI(int wages, int interest, int unemployment) {
8     /* Complete the function and update the return statement */
9
10    return -1;
11 }
12
13 // Calculate deduction depending on single, dependent or married
14 int GetDeduction(int status) {
15     /* Complete the function and update the return statement */
16 }
```

Develop mode**Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)

**main.cpp**

(Your program) 4 18:14 1939727

Rob Daglio

MDCCOP2335Spring2024

Program output displayed hereCoding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 01/31/24 18:14 1939727

Rob Daglio
MDCCOP2335Spring2024

©zyBooks 01/31/24 18:14 1939727

Rob Daglio
MDCCOP2335Spring2024