

ACMS 20210, Final Exam

This exam consists of six problems, each weighted equally. Justify your responses and show all work, unless indicated otherwise. All work submitted is to be your own, and compliance with the honor code is mandatory.

- You do not have to account for checking whether inputs are correct (i.e. always assume the user enters a reasonable value).
- For writing functions, you may assume that all vectors/maps/etc. passed to the functions that you write have size at least 1.
- For writing functions, you may assume that the program contains the appropriate includes, declarations, and using statements before the definition of your function.
- For writing programs, you must write out the appropriate includes, declarations, and using statements.

1. Write a C++ program that accepts three command line arguments: an integer a , an integer b , and an integer N . The program then draws N integers uniformly at random from the uniform integer distribution on $[a..b]$ (the integers between a and b inclusive) and reports the number of times each integer was drawn back to the user. (Hint: use a map or vector.)

2. Write a C++ program that prompts the user to enter three values: an integer a , an integer b , and a positive integer n . It then calculates the sum of all of the integers in the interval $[a, b]$ that are not divisible by n , and reports this value to the user.

3. Write a short C++ function that accepts a map from strings to integers as an input, and returns a key (string) that has the largest associated value (integer). You may assume there will be only one key with the largest value for this function (or that it does not matter which key with the largest value that you will return).

4. Write a short C++ function that accepts two positive integers, m and n (you may assume $n \geq m$), as inputs and returns a double representing the quantity:

$$S = \sum_{i=m}^n \left(\sum_{j=i}^n \frac{i^2}{j} \right)$$

5. Write a short C++ function that accepts a positive integer x and a vector of doubles v as inputs and returns the sum of the reciprocals of the entries of v that are greater in absolute value than x , i.e.

$$S = \sum_{|v_i| > x} \frac{1}{v_i}$$

6. Write a short C++ function that accepts two strings as inputs. If the strings have the same length, the function returns a string made by interweaving the strings, alternating letters from each of the inputs. For example, if the two strings passed to the function were “circle” and “square”, the string returned by the function would be “csiqrucalree”. If the strings passed in have different lengths, the function returns an empty string.

7. Write a short C++ function that accepts a `vector<vector<double>>` as input, representing a matrix. The matrix represented is guaranteed to be square (have the same number of rows and columns). The function then computes the trace of the matrix (sum of the entries along the main diagonal).

8. Write a C++ program that will read a file called “transactions.dat” containing an unknown number of lines consisting of a first name followed by a space followed by an integer (either positive or negative) representing a transaction that the person makes. Each person may make multiple transactions, and each person’s balance starts at 0 at the beginning of the program, and no two people have the same name. After processing all of the transactions, write to each line the name of a person, followed by a space, followed by the balance of that person, in alphabetical order, in a file called “balances.dat”.

The contents of the file might look like the following:

John 100

Mary 200

George 50

John -40

Mary 100

Mary -50

George 200

John 30

9. Write a C++ program that computes the definite integral of a function $f(x)$ (defined in the separate header file called `function_to_integrate.h`) over an interval $[a, b]$ using the trapezoidal rule. The program should take three command line arguments: a double representing a , a double representing b , and a positive integer N representing the number of subintervals.

10. Write a short recursive C++ function that computes $f(n)$ for a positive integer n , where

$$f(n) = \begin{cases} 1, & \text{for } n = 1 \\ 2, & \text{for } n = 2 \\ f(n-1) + f(\frac{n+1}{2}), & \text{for } n \text{ odd otherwise} \\ f(\frac{n}{2}) + f(\frac{n-2}{2}), & \text{for } n \text{ even otherwise} \end{cases}$$

Additionally, calculate $f(9)$ by hand.

11. In mathematical logic, there are two additional important logical implications that we did not discuss in class: implications ($P \implies Q$, read “ P implies Q ”) and biconditionals ($P \iff Q$, read “ P if and only if Q ”).

Here is the truth table for $P \implies Q$:

P	Q	$P \implies Q$
T	T	T
T	F	F
F	T	T
F	F	T

Here is the truth table for $P \iff Q$:

P	Q	$P \iff Q$
T	T	T
T	F	F
F	T	F
F	F	T

- Verify the following: $\sim (P \implies Q) \equiv P \wedge \sim Q$

- Verify the following: $(P \implies Q) \wedge (Q \implies P) \equiv P \iff Q$

- Verify the following: $P \implies (Q \vee R) \equiv (P \implies Q) \vee (P \implies R)$

- Verify the following: $P \wedge (\sim (P \vee Q))$ is a contradiction.

- Verify the following: $P \wedge (\sim (P \wedge Q)) \equiv (P \wedge \sim Q)$

12. Write a C++ program to perform a Monte Carlo simulation to approximate the expected number of steps that a simple symmetric random walk in \mathbb{Z}^2 (starting from the origin, where each step is north, east, south, or west with equal probability) will take until it is a distance of R or more from the origin. Your program should take two command line arguments: a double representing the distance R and a positive integer representing the number of trials. Report the approximate expected number of steps to the user.

13. A particle moves forward along a line, starting from the origin, until it meets or passes a certain position F , which we will call the finish line. At each step, the particle advances a random distance forward, selected from the uniform (real) distribution on $[0, M)$. Write a program using a Monte Carlo simulation to approximate the expected number of steps required until the particle meets or passes the finish line. Your program should take three command line arguments: a double representing the finish line F , a double representing the maximum step size M , and an integer representing the number of trials that your simulation will run. Report the approximate expected number of steps required to the user.

- 14.
- Give at least one reason for using classes in programming.
 - Explain the difference between pass by value, pass by reference, and pass by constant reference in C++.
 - Give a brief description of some of the differences between the map and vector container classes in C++.
 - In the following code, how many arithmetic operations (+, -, *, /) are performed?

```
int total = 0;
for(int i = 0; i < 5; i++){
    for(int j = i; j < 5; j++){
        total += 2 * i + j;
    }
    total += 1;
}
```
 - Why is it dangerous to check for equality using == in floating point arithmetic?