NYU, Tandon School of Engineering
CS Bridge — Summer 2023

Exam 2

Thursday, August 31, 2023

- This exam has 6 questions, with 100 points total.

- **You should submit your answers in the Gradescope platform (not on NYU Brightspace).**

- You have two hours.

- **It is your responsibility to take the time for the exam** (You may use a physical timer, or an online timer: https://vclock.com/set-timer-for-2-hours/).
  Make sure to upload the files with your answers to gradescope BEFORE the time is up, while still being monitored by ProctorU.
  We will not accept any late submissions.

- In total, you should upload 3 '.cpp' files:
  - One '.cpp' file for questions 1-4.
    Write your answer as one long comment (/* … */).
    Name this file 'YourNetID_q1to4.cpp'.
  - One '.cpp' file for question 5, containing your code.
    Name this file 'YourNetID_q5.cpp'.
  - One '.cpp' file for question 6, containing your code.
    Name this file 'YourNetID_q6.cpp'.

- **Write your name, and netID at the head of each file.**

- This is a closed-book exam. However, you are allowed to use:
  - Visual-Studio, Visual Studio Code (VSCode), Xcode, CLion. You should create a new project and work ONLY in it.
  - Two sheets of scratch paper.
  - Scientific Calculator (Physical or Operating System's Provided One).

  Besides that, no additional resources (of any form) are allowed.

- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.

- Read every question completely before answering it.
  Note that there are 2 programming problems at the end.
  Be sure to allow enough time for these questions

Part I – Theoretical:

- *You should submit your answers to all questions in this part (questions 1-4) in* ***one*** *'.cpp' file. Write your answers as one long comment (`/* … */`).*
  Name this file 'YourNetID_q1to4.cpp'.

- *For questions in this part, try to find a way to use regular symbols.*
  For example, instead of writing a^b you could write a^b, instead of writing $\Theta(n)$, you could write theta(n), instead of writing $\binom{n}{k}$ you could write C(n, k), etc.
  Alternatively, you could also make a note, at the beginning of your answer, stating what symbol you used to indicate a specific mathematical notation.

Question 1 (13 points)
Use mathematical induction to prove that 5 divides $7^n - 2^n$ whenever n is a non-negative integer, that is, $n \geq 0$.

Question 2 (16 points)

a) In a technician's box there are 400 VLSI chips, 12 of which are faulty. How many ways are there to pick two chips, so that one is a working chip and the other is faulty? (Assume that no chips are identical.) **Explain your answer.**

b) Find the number of subsets of **S = {1, 2, 3, . . . , 10}** that contain exactly four elements, the sum of which is even. **Explain your answer.**

Question 3 (18 points)

a) A red and a green die are rolled. What is the probability of getting a sum of six, given that the number on the green die is odd? **Explain your answer.**

b) Suppose you flip a biased coin, where the probability of getting head is $\frac{2}{3}$. and the probability of getting tail is $\frac{1}{3}$, 6 times. What is the probability of getting at most 2 heads out of these 6 flips? **Explain your answer.**

Question 4 (18 points)
Analyze its running time of `function1` and `function2`.
Explain your answers.

Note: Give your answers in terms of asymptotic order. That is, $T(n) = \Theta(n^2)$, or $T(n) = \Theta(\sqrt{n})$, etc.

```
int function1(int n){
    int i, j;
    int sum = 0;

    i = 0;
    while (i <= n/2){
          sum = sum + 1;
          i++;
    }

    sum = 0;
    for (i = 1; i <= n; i *= 5)
        for (j = 1; j <= i; j++)
            sum += (i+j);

    for (i = 1; i <= n; i *= 5)
        for (j = 1; j <= n; j++)
            sum += (i+j);

    return sum;
}


int function2(int n){
    int i, j;
    int sum = 0;
    if (n == 500){
      i = 0;
      while (i <= n) {
           sum = sum + 2;
           i = i + 1;
      }
    }
    else{
        for (i = 1; i <= n; i *= 5){
            j = i;
            while (j > 1){
                sum += 1;
                j /= 5;
            }
        }
   }
    return sum;
}
```

Part II – Coding:

- *Each question in this part (questions 5-6), should be submitted as a '.cpp' file.*
- *Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.*
- *In all questions, you may assume that the user enters inputs as they are asked. For example, if the program expects a positive integer, you may assume that user will enter positive integers.*
- *No need to document your code. However, you may add comments if you think they are needed for clarity.*

Question 5 (18 points)

Give a **recursive** C++ implementation for the function:

```
int Count_Sum_Primes(int start, int end, int &countPrimes)
```

The above function is given an integer **start,** another integer **end,** and an address to an integer variable **countPrimes**. When this `Count_Sum_Primes` function is called, it should **return the sum of all the prime numbers within the range [start, end]** and updates the parameter **countPrimes with the value of the total number of prime numbers within the range [start, end]**.

For example, after calling `Count_Sum_Primes (2, 7, countPrimes)`, this function should return **17** and value of parameter **countPrimes** should be **4**.

For example, after calling `Count_Sum_Primes (4, 17, countPrimes)`, this function should return **53** and value of parameter **countPrimes** should be **5**.

For example, after calling `Count_Sum_Primes (6, 15, countPrimes)`, this function should return **31** and value of parameter **countPrimes** should be **3**.

Implementation requirements:

- Your function **must be recursive**.
- You can assume that initial value of **countPrimes** integer variable is Zero (0).
- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.
- If you use a function to determine whether a number is prime or not, that function could be iterative or recursive. But `Count_Sum_Primes function must be recursive.`

Note: You don't need to write a `main()` function.

Question 6 (17 points):
Give a C++ implementation for the function:

```
int Complete_Vector(vector<int> Vector);
```

The above function is given an integer vector **Vector** (**type vector <int>**) that will contain **unique integers** in the **range [0, n]** where **n** is the size of the **Vector**. Notice that, there is **one number** from this **range [0, n]** that is not in **Vector**. When this `Complete_Vector` function is called, it should **find and return the missing integer**.

For example, if **vector <int> Vector `{5, 6, 0, 1, 2, 3},`** after calling `Complete_Vector(Vector) function`, it should return 4.

For example, if **vector <int> Vector `{4, 5, 1, 2, 3},`** after calling `Complete_Vector(Vector) function`, it should return `0`.

For example, if **vector <int> Vector `{4, 5, 1, 2, 3, 0, 6, 7, 8},`** after calling `Complete_Vector(Vector) function`, it should return 9.

Implementation requirements:

- Your function should run in $\theta(n)$ where n = size of the **Vector** (**type vector <int>**). For Simplicity, you can assume that amortized $\theta(1)$ is same as $\theta(1)$**.**
- Your function should use only $\theta(1)$ additional memory, that is, you are not allowed to use extra memory which is dependent on size of the input vectors. You may use some additional variables and/or some fixed length vectors or array.
- You should not modify the input vector **Vector** (**type vector <int>**).
- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.

Note: You don't need to write a `main()` function.