

Exam 2

Thursday, March 23, 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'.
- You must show the numerical expression and/or arithmetic equation in unreduced form that produces the final answer.
- 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 $\text{theta}(n)$, 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 for all integers $n \geq 1$,
 $(1 + 2^n) \leq 3^n$.

Question 2 (16 points)

- Each user has a password 5 characters long where each character is an uppercase letter, a lowercase letter, or a digit. Each password must contain at least one digit. How long will it take to check every possible character combination if each check takes one unit of time. **Explain your answer.** You don't need to reduce the numerical expression to a numeric answer.
- Find the number of subsets of $S = \{1, 2, 3, \dots, 10\}$ that contain exactly five elements, but neither 3 nor 4. **Explain your answer.**

Question 3 (18 points)

- Suppose you pick a bit string from the set of all bit strings of length eight. What is the probability that the bit string has more 0's than 1's? **Explain your answer.**
- A pair of dice, each with the numbers 1, 1, 2, 2, 2, 3 on its six sides are rolled. What is the expected value of the sum of the numbers showing? **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, k;
    int sum = 0;

    for (i = 1; i <= 100; i *= 2)
        for (j = 1; j <= n; j *= 2){
            k = j;
            while (k > 1){
                sum += (i+j);
                k /= 2;
            }
        }

    for (i = 1; i <= n; i *= 2)
        sum += i;

    return sum;
}

int function2(int a, int n){ //where a >= 1 and n >= 1
    if(n == 1)
        return a;

    else{
        if(n%2 == 0)
            return function2(a, n/2) * function2(a, n/2);
        else
            return a * function2(a, (n-1)/2) * function2(a, (n-1)/2);
    }
}
```

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 (15 points)

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

```
int countConsonants(string S, int length)
```

The above function is given a **non-empty** string **S** that will contain **alphanumeric characters ([A-z, 0-9])**, and an integer **length** that will indicate the **logical size** of the string **S**. When this `countConsonants` function is called, it should **return the total number of consonants** in string **S**.

For example, if **S = "CSBridge"**, after calling `countConsonants (S, 8)`, this function should return **6**.

For example, if **S = "todayissunny"**, after calling `countConsonants (S, 12)`, this function should return **8**.

For example, if **S = "PathwaytoCSTandonMS"**, after calling `countConsonants (S, 19)`, this function should return **14**.

Implementation requirements:

- Your function should run in **worst case linear time**. That is, it should run in $\theta(n)$ where n = logical size of the string **S**.
- Your function **must be recursive**.
- You should not use any iterative statements (for/while/do-while loop) while implementing the above function.
- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.
- **Alphanumeric characters** mean lowercase English letters ([a-z]), uppercase English letters ([A-Z]), and numerical characters ([0-9]).
- **Consonants** means all the uppercase English letters except (A, E, I, O, U) and all the lowercase English letters except (a, e, i, o, u).

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

Question 6 (20 points):

In this question, you should write a program that reads a sequence of strings (each string will consist of lowercase English letters (**[a-z]**) and/or numerical characters (**[0-9]**)) and removes all the strings that consist of only lowercase English letters and then prints the strings (after removal of strings that consist of only lowercase English letters) according to the insertion order. At the end, you should also print how many of the input strings contain only lowercase English letters. That is, the program will remove input strings that contain only lowercase English letters and print those input strings that contain at least one numerical character according to the insertion orders, and then print how many of the input strings contain only lowercase English letters.

The input would be entered as a non-empty sequence of lines, where each line would contain a single string, and an empty string will indicate the end of the input.

After reading the input, the program would remove strings that contain only lowercase English letters and then print those strings that contain at least one numerical character maintaining the insertion order, followed by a number that will indicate how many strings in the input sequence contain only lowercase English letters. Your program should ignore the inputted empty string that was used to indicate the end of input.

Your program should interact with the user **exactly**, as demonstrated below:

Example 1:

Please enter a non-empty sequence of Strings. Each string should be in a separate line and consists of only lowercase English letters and/or numerical characters. To indicate the end of the input, enter an empty string in one line.

```
abc123defghi
abcdef
sfja123
kjm
xyz123pqr
456uyt
```

```
abc123defghi
sfja123
xyz123pqr
456uyt
```

```
Number of Strings in the input sequence that contain only lowercase
English letters: 2
```

Example 2:

Please enter a non-empty sequence of Strings. Each string should be in a separate line and consists of only lowercase English letters and/or numerical characters. To indicate the end of the input, enter an empty string in one line.

906abcd

zuequ

csbridge

csbridge123

9999

abcxyz

mkjhyuioperdswqa

abcdeftqtyqraf

12345678

efg678

906abcd

csbridge123

9999

12345678

efg678

Number of Strings in the input sequence that contain only lowercase English letters: 5

Notes:

1. Your program should ignore the inputted empty string that was used to indicate the end of input.
2. Make sure to **design your program best**. In particular, break your implementation to functions.
3. You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.