Exam 2

Thursday, September 21, 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
$1 + 3 + 9 + 27 + \cdots + 3^n = (3^{n+1} - 1)/2$, whenever n is a non-negative integer, that is, $n \geq 0$.

Question 2 (16 points)

a) Suppose there are nine people (Ann, Ben, Cal, Dot, Ed, Fran, Gail, Hal, and Ida) are in a room. Five of them stand in a row for a picture. In how many ways can this be done if Hal or Ida (but not both) are in the picture? **Explain your answer.**

b) Find the number of subsets of **S = {1, 2, 3, . . . , 10}** with exactly five elements, two of which are 3 and 4. **Explain your answer.**

Question 3 (18 points)

a) Suppose you have a class with 30 students — 10 freshmen, 12 sophomores, and 8 juniors. You pick two students at random, one at a time. What is the probability that the second student is a freshman, given that the first is a freshman? **Explain your answer.**

b) 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.**

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 total = 0;


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


    if(n%5 == 0){
        for (i = 1; i <= n; i++)
            for (j = 1; j <= n; j += 1)
                total += (i+j);
    }

    return total;
}


int function2(int n){
    int i, j;
    int total = 0;

    for (i = 1; i <= n; i *= 5){
        j = i;
        while (j > 1){
            total += 1;
            j /= 5;
        }
    }


    return total;
}
```

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 (17 points)

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

```
int findOddIntegersAndSum(int S[], int length, int &sumOdds)
```

The above function is given an integer array **S** that will contain **integers,** an integer **length** that will indicate the **logical size** of the array **S,** and an address to an integer variable **sumOdds**. When this `findOddIntegersAndSum` function is called, it should **return the total count of the odd integers** appears in array S and updates the parameter **sumOdds with the total summation of all the odd integers in array S**.

For example, if **S = {10, –5, –20, 0, 7, –15, –5, –100}**, after calling `findOddIntegersAndSum(S, 8, sumOdds)`, this function should return **4** and value of parameter **sumOdds** should be **-18.**

For example, if **S = {–1, –5, –70, –15, –26, –44, 0, 25, –14, –3}**, after calling `findOddIntegersAndSum(S, 10, sumOdds)`, this function should return **5** and value of parameter **sumOdds** should be **1.**

For example, if **S = {9, 0, 13, 17, 29, –14, –60, 9, –9}**, after calling `findOddIntegersAndSum(S, 9, sumOdds)`, this function should return **6** and value of parameter **sumOdds** should be **68.**

For example, if **S = {1, 5, 20, 0, –40}**, after calling `findOddIntegersAndSum(S, 5, sumOdds)`, this function should return **2** and value of parameter **sumOdds** should be **6.**

Implementation requirements:

- Your function should run in **worst case linear time**. That is, it should run in $\boldsymbol{\theta(n)}$ where n = logical size of the array S.
- Your function **must be recursive**.
- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.
- You can assume that initial value of **sumOdds** integer variable is Zero (0).
- You can assume that zero (0) as an even number.

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

Question 6 (18 points):

In this question, you should write a program that reads a sequence of strings (each string will consist of only decimal digit characters) and removes the strings which contain at least one decimal digit character other than character '0' or '1' and then prints the strings which are consisting of only characters '0' and/or '1' according to the insertion order. At the end, you should also print how many strings got deleted/removed because each of the those deleted strings contain at least one decimal character other than '0' or '1'. That is, the program will remove input strings which contain at least one of the decimal digit characters other than '0' or '1' and then print those strings which are consisting of only '0' and/or '1' according to the insertion orders, and then print how many of the input strings get deleted because of containing at least one decimal digit character other than '0' or '1'.

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

After reading the input, the program would remove the strings that contain at least one decimal digit character which is not '0' or '1' and then print those strings that are consisting of only decimal digit characters '0' and/or '1' maintaining the insertion order, followed by a number that will indicate how many strings in the input sequence get deleted. 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 decimal digit
characters. To indicate the end of the input, enter an empty
string in one line.
10234
123456
01011
09827
101011
76510981
0
1
12301
00

01011
101011
0
1
00
Number of Strings in the input sequence that get deleted for containing
at least one character which is not 0 or 1:  5
```

Example 2:

```
Please enter a non-empty sequence of Strings. Each string should
be in a separate line and consists of only decimal digit
characters. To indicate the end of the input, enter an empty
string in one line.
765
123
98
01
10
1
0000
98765432
83723
011100
96785
0000
98981
45
11
3
000

01
10
1
0000
011100
0000
11
000
Number of Strings in the input sequence that get deleted for
containing at least one character which is not 0 or 1:  9
```

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.
4. Decimal digit characters are ‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, and ‘9’.