NYU, Tandon School of Engineering
CS Bridge — Spring 2022

Exam 2

Thursday, May 26, 2022

- 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. 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 for every positive integer n, $(4^{n+1} + 5^{2n-1})$ is divisible by 21.

Question 2 (16 points)

Nine women and eleven men are on the faculty in the mathematics department at a school.

a) How many ways are there to select a committee of five members of the department if at least one woman must be on the committee?
Explain your answers.

b) How many ways are there to select a committee of five members of the department if at least one woman and at least one man must be on the committee?
Explain your answers.

Question 3 (18 points)

Suppose that we roll a fair die until a 4 comes up.

a) What is the probability that we roll the die n times?
Explain your answers.

b) What is the expected number of times we roll the die?
Explain your answers.

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;

    if(n%3 == 0){
        for (i = 1; i <= n; i *= 3)
            for (j = 1; j <= n; j += 1)
                total += (i+j);
    }

    if(n%2 == 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 *= 2){
        j = i;
        while (j > 1){
            total += 1;
            j /= 2;
        }
    }


    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 (15 points)

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

```
bool isSorted(int S[], int length)
```

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**. When this `isSorted` function is called, it should **return true** if the elements of array S are sorted in non-increasing order. Otherwise, it should **return false**.

For example, if **`S = {100, 75, 27, 15, 8, -5, -5, -10}`**, after calling `isSorted(S, 8)`, this function should return **true**. Because **array S** is sorted in non-increasing order.

For example, if **`S = {-1, -5, -7, -15, -26, -44, 0, 5, -1, -3}`**, after calling `isSorted(S, 10)`, this function should return **false**. Because **array S** is not sorted in non-increasing order.

For example, if **`S = {1, 15, 81, 199, 300, 400}`**, after calling. `isSorted(S, 6)`, this function should return **false**. Because **array S** is not sorted in non-increasing order.

For example, if **`S = {69, 50, 33, 25, 23, 14, 5, 3, 0, -7, -9}`**, after calling `isSorted(S, 11)`, this function should return **true**. Because **array S** is sorted in non-increasing order.

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.

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 alphanumeric characters) and removes letters/alphabets from each string and then prints the strings (after removal of letters/alphabets) according to the insertion order. If a string becomes empty after removing the letters/alphabets, you don't need to print that string. At the end, you should print how many strings have become empty strings after removal of letters/alphabets. That is, the program will remove letters/alphabets from each of the input strings and print those strings according to the insertion orders, and then print how many of the input strings become empty after removal of letters/alphabets.

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 alphanumeric characters), and a empty string will indicate the end of the input.

After reading the input, the program would remove letters/alphabets from each of the strings and then print those strings (after removal of letters/alphabets) maintaining the insertion order, followed by a number that will indicate how many strings in the input sequence were consisted of only letters/alphabets. If a string in the input sequence becomes empty after removal of letters/alphabets, you don't need to print that empty string. 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 alphanumeric
characters. To indicate the end of the input, enter an empty
string in one line.
10ashfasowshvau29ueq9u233
Sakhfkahka234ABB436
123456
ahfakhkakankbkhBBBaaa
ABCXYZ123abcxyz09827
985aljlaj
afkabkaf9
638oioja789

10299233
234436
123456
09827
985
9
638789
Number of Strings in the input sequence that contain no numeric
characters:  1
```

Example 2:

```
Please enter a non-empty sequence of Strings. Each string should
be in a separate line and consists of only alphanumeric
characters. To indicate the end of the input, enter an empty
string in one line.
Ashfasowshvau0910akhdka29ueq9u1267
aadjlsjlanalJLfkah1ka892ABB123
98khkhay3728kanakkhad
aathgaoouylajdgag
1837jnajojaouqBVGpjlaj123
11113ha34522
aojdaojaljaABCXYZ523abcxyz091827
96785
abheyoqnayqKUUuwyywwqe
Aadjalj0000
98q9euoa1981
87

09102991267
892123
983728
1837123
1111334522
523091827
96745
0000
9891981
87
Number of Strings in the input sequence which contain no numeric
characters:  2
```

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.