NYU, Tandon School of Engineering
Extended Bridge to CS — Summer 2021

Homework #7
Due by Friday 8/27 11:59pm

Submission instructions:

1. For this assignment, you should turn in 5 files:
   - Two '.cpp' files, one for each question 1-2.
     Name your files 'YourNetID_hw7_q1.cpp', and 'YourNetID_hw7_q2.cpp'.
   - One '.pdf' file with your answers for questions 3-7.
     Each question should start on a new page!
     Name your file 'YourNetID_hw7_q3to7.pdf'

2. You must type all your solutions. We will take off points for submissions that are handwritten.

3. **You should submit your homework in the Gradescope system.** Note that when submitting the pdf file, you would be asked to assign the pages from your file to their corresponding questions.

4. **You can work and submit in groups of up to 4 people. If submitting as a group, make sure to associate all group members to the submission on gradescope.**

5. Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose the most appropriate control flow statements, break down your solutions by defining functions, etc.

6. For the math questions, you are expected to justify all your answers, not just to give the final answer (unless explicitly asked to).
   As a rule of thumb, for questions taken from zyBooks, the format of your answers, should be like the format demonstrated in the sample solutions we exposed.

Question 1:

a. Implement a function:

`int printMonthCalender(int numOfDays, int startingDay)` This function is given two parameters:

- `numOfDays` - The number of days in the month
- `startingDay` – a number 1-7 that represents the day in the week of the first day in that month (1 for Monday, 2 for Tuesday, 3 for Wednesday, etc.).

The function should:

- Print a formatted monthly calendar of that month

- Return a number 1-7 that represents the day in the week of the **last day** in that month.

Formatting Notes:

- The output should include a header line with the days' names.
- Columns should be spaced by a Tab.

Example: when calling `printMonthCalender(31, 4)` it should return 6, and should print:

```
Mon Tue Wed Thr Fri Sat Sun
                          1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

b. A method for determining if a year is a leap year in the Gregorian calendar system is to check if it is divisible by 4 but not by 100, unless it is also divisible by 400. For example, 1896, 1904, and 2000 were leap years but 1900 was not. Write a function that takes in a year as input and return true if the year is a leap year, return false otherwise.

Note: background on leap year https://en.wikipedia.org/wiki/Leap_year

c. Implement a function:

`void printYearCalender(int year, int startingDay)`

This function is given two parameters:

- `year` – an integer that represents a year (e.g. 2016)
- `startingDay` – a number 1-7 that represents the day in the week of 1/1 in that year (1 for Monday, 2 for Tuesday, 3 for Wednesday, etc.).

The function should use the functions from sections (a) and (b) in order to print a formatted yearly calendar of that year.

Formatting Note: As the header for each month you should print the months' name followed by the year (e.g. March 2016).

Example: Appendix A shows the expected output of the call `printYearCalender(2016, 5).`

d. Write program that interacts with the user and your function in (c).

Question 2:

Consider the following definitions:

a. A **proper divisors** of a positive integer (≥ 2) is any of its divisors excluding the number itself.

For example, the proper divisors of 10 are: 1, 2 and 5.

b. A **perfect number** is a positive integer (≥ 2) that is equal to the sum of its proper divisors.

For example, 6 and 28 are perfect numbers, since:

6 = 1 + 2 + 3

28 = 1 + 2 + 4 + 7 + 14

Background of perfect numbers: https://en.wikipedia.org/wiki/Perfect_number c.
Amicable numbers are two different positive integer (≥ 2), so related that the sum of the proper divisors of each is equal to the other number.

For example, 220 and 284 are amicable numbers, since:

284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110

220)

220 = 1 + 2 + 4 + 71 + 142

(proper

(proper divisors of

divisors of 284)

Background of amicable numbers: https://en.wikipedia.org/wiki/Amicable_numbers

a. Write a function:

```
void analyzeDividors(int num, int& outCountDivs, int& outSumDivs)
```

The function takes as an input a positive integer `num` (≥ 2), and updates two output parameters with the number of `num`'s proper divisors and their sum.

For example, if this function is called with `num`=12, since 1, 2, 3, 4 and 6 are 12s proper divisors, the function would update the output parameters with the numbers 5 and 16.

<u>Note</u>: Pay attention to the running time of your function. An efficient implementation would run in Θ"√���'.

b. Use the function you wrote in section (a), to implement the function:

```
bool isPerfect(int num)
```

This functions is given positive integer `num` (≥ 2), and determines if it is perfect number or not.

c. Use the functions you implemented in sections (a) and (b), to write a program that reads from the user a positive integer M (≥ 2), and prints:

- All the perfect numbers between 2 and M.
- All pairs of amicable numbers that are between 2 and M (both numbers must be in the range).

<u>Note</u>: Pay attention to the running time of your implementation. An efficient algorithm for this part would call `analyzeDividors` Θ(�) times all together.

Question 3:

a.）Solve Exercise 8.2.2, section b from the Discrete Math zyBook.

b.Solve Exercise 8.3.5, sections a-e from the Discrete Math zyBook

$$f(n) = n^3 + 3n^2 + 4 \cdot f = \theta(n^3)$$

So: Let c=8 and $n_0 = 1$, then for any n $\geq 3n_0$,

$f(n) = n^3 + 3n^2 + 4 \cdot f = \theta(n^3)$

So: Let c=8 and $n_0 = 1$, then for any n $\geq 3n_0$,

$$n^3 + 3n^2 + 4 \leq n^3 + 3n^3 + 4n^3$$

$$= 8 * n^3$$

So, $f(n) = \Omega(n^3)$

b) Exercise 8.3.5 a-e

(a)

Describe in English how the sequence of numbers is changed by the algorithm. (Hint: try the algorithm out on a small list of positive and negative numbers with p = 0)

The outer loop, keep iterate when i<j, inner loop keeps iterate when i<j.

First while end with i+1=n, so i =n;

Second while end with n-1=2, so n=1

So n and i swap.

(b)

What is the total number of times that the lines "i := i + 1" or "j := j - 1" are executed on a sequence of length n? Does your answer depend on the actual values of the numbers in the sequence or just the length of the sequence? If so, describe the inputs that maximize and minimize the number of times the two lines are executed.

Total times depends on the value. Based on P.

Max: all negative in sequence and p=0 , iterate to the end.

Min: all positive and p=0, and min=0.

(c)

What is the total number of times that the swap operation is executed? Does your answer depend on the actual values of the numbers in the sequence or just the length of the sequence? If so, describe the inputs that maximize and minimize the number of times the swap is executed.

Depend on the value of number in sequence.

Max:the times i will increment, or j decremente

Min: if p=the actual number, and all the listed number is bigger than it

---

(d)

Give an asymptotic lower bound for the time complexity of the algorithm. Is it important to consider the worst-case input in determining an asymptotic lower bound (using Ω) on the time complexity of the algorithm? (Hint: argue that the number of swaps is at most the number of times that i is incremented or j is decremented).

The inner while loop will always be n.

Counter keep executed n times. The loop executed until i<j.

So time complexity is n times. $so\ it\ is\ \Omega(n)$.

(e)

Give a matching upper bound (using O-notation) for the time complexity of the algorithm.

Matching upper bound is as O(n).

Question 4:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.1.2, sections **b, c**

- Digits = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
- Letters = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z }
- Special characters = { *, &, $, # }

(b) Strings of length 7, 8, or 9. Characters can be special characters, digits, or letters.

Let L be the set of all lower case letters ,D be the set of digits and S be the set of all special characters. .|L| = 26 and |D| = 10, |S| = 4.

The set of all allowed characters is C = L ∪ D ∪ S. Since D ∩ L∩S = ∅, the sum rule can be applied to find the cardinality of C: |C| = 26 + 10 + 4 = 40.

Let A_j denote the strings of length j over the alphabet C. By the product rule, $|A_j| = 40_j$.

Notice that for j ≠ k, A_j and A_k are disjoint because a string can not have length j and length k at the same time.

If the user must select a password of length 7, 8, or 9, then the sum rule applies:

$$|A_7 \cup A_8 \cup A_9| = |A_7| + |A_8| + |A_9| = 40^7+40^8+40^9$$

(c)Strings of length 7, 8, or 9. Characters can be special characters, digits, or letters. The first character cannot be a letter.

Let L be the set of all lower case letters ,D be the set of digits and S be the set of all special characters. .|L| = 26 and |D| = 10, |S| = 4.

The set of all allowed characters is C = L ∪ D ∪ S. Since D ∩ L∩S = ∅, the sum rule can be applied to find the cardinality of C: |C| = 26 + 10 + 4 = 40.

Let A_j denote the strings of length j over the alphabet C. By the product rule, $|A_j| = 40_j$.

Notice that for j ≠ k, A_j and A_k are disjoint because a string can not have length j and length k at the same time.

Define S to be the set of binary strings of length 7 that start and end with|F|=|D|+ |S|. Since the first character cannot be a letter. In this condition , F: |F| = 10 + 4 = 14.

A string is in the set S if it has the form |F|******, where each * could be a |L| + D| + |S|=|C|. $|A_7|=14*40^6$, etc.

If the user must select a password of length 7, 8, or 9, then the sum rule applies:

$$|A_7 \cup A_8 \cup A_9| = |A_7| + |A_8| + |A_9| =14*40^6+14* 40^7+14 * 40^8=14*(40^7+40^8+40^6)$$

b) Exercise 5.3.2, section a

(a)

How many strings are there over the set {a, b, c} that have length 10 in which no two consecutive characters are the same? For example, the string "abcbcbabcb" would count and the strings "abbbcbabcb" and "aacbcbabcb" would not count.

$3 * 2^9 = 1536$

---

c) Exercise 5.3.3, sections b, c

(b)

How many license plate numbers are possible if no digit appears more than once?

$10 * 26^4 * 9 * 8$

(c)

How many license plate numbers are possible if no digit or letter appears more than once?

$10 * 9 * 8 * 26 * 25 * 24 * 23$

---

d) Exercise 5.2.3, sections a, b

Let B = {0, 1}. B^n is the set of binary strings with n bits.

Define the set E_n to be the set of binary strings with n bits that have an even number of 1's. Note that zero is an even number, so a string with zero 1's (i.e., a string that is all 0's) has an even number of 1's.

(a)Show a bijection between B^9 and E10. Explain why your function is a bijection.

A function f is a bijection from the power set of B = {0, 1} to the set of 9-bit strings.

f: B^9= $|\{0, 1\}^9|$.=2^9, the $\varnothing$ does not include 0, or 1, therefore f(∅)=000000000.

The function is from set B to set E , like this $f: |B^9| \rightarrow |E10|$, when x ∈ B^9, then x is counted by collecting 1's in the string.

if number of 1 in B^9 is odd, then we add 1 as one more bit in E10.

f: 000000001 → 0000000011.

If number of 1 in B^9 is even, then we add 0 as one more bit in E10.

f: 000000011 → 00000000110.

$|B^9| = 2^9$

$|E10| = (|B^9|/2) * 2 = |B^9| = (2^9/2) * 2 = 2^9$

f is one to one is because x and y ∈ B^9, f: $|B^9| \rightarrow |E10|$= g: $|E10| \rightarrow |B^9|$, 2^9=2^9;

f is onto is because every y∈ $E10$, $there's\ exactly\ one\ element\ in\ the\ B,\ \ the\ first\ 9\ bits,$ x ∈ B^9, f(x)=y, if only if g(y)=x. F has an inverse f-1.

So, f is a bijection.

(b) What is |E10|?

$|E10| = (|B^9|/2) * 2 = |B^9| = (2^9/2) * 2 = 2^9$

Question 5:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.4.2, sections a, b

At a certain university in the U.S., all phone numbers are 7-digits long and start with either 824 or 825.

(a)

How many different phone numbers are possible?

$2 * 10^4$

(b)

How many different phone numbers are there in which the last four digits are all different?

$2 * 10 * 9 * 8 * 7$

---

b) Exercise 5.5.3, sections a-g

How many 10-bit strings are there subject to each of the following restrictions?

(a)No restrictions.= 2^{10}

(b)The string starts with 001.= 2^7

(c)The string starts with 001 or 10.= 2^7+2^8

(d)The first two bits are the same as the last two bits.= 2^8

(e)The string has exactly six 0's.= $\binom{10}{6}$

(f)The string has exactly six 0's and the first bit is 1.= $\binom{9}{6}$

(g)There is exactly one 1 in the first half and exactly three 1's in the second half. =
$\binom{5}{1} * \binom{5}{3}$

---

c) Exercise 5.5.5, section a

(a)

There are 30 boys and 35 girls that try out for a chorus. The choir director will select 10 girls and 10 boys from the children trying out. How many ways are there for the choir director to make his selection?

$$\binom{30}{10}\binom{35}{10}$$

---

d) Exercise 5.5.8, sections c-f

(c)

How many five-card hands are made entirely of hearts and diamonds?

$$\binom{26}{5} = 65780$$

(d)

How many five-card hands have four cards of the same rank?

$$\binom{13}{1}\binom{48}{1} = 624$$

(e)

A "full house" is a five-card hand that has two cards of the same rank and three cards of the same rank. For example, {queen of hearts, queen of spades, 8 of diamonds, 8 of spades, 8 of clubs}. How many five-card hands contain a full house?

$$\binom{13}{1}\binom{4}{3}\binom{12}{1}\binom{4}{2} = 3744$$

(f)

How many five-card hands do not have any two cards of the same rank?

$$\binom{13}{5} * 4^5$$

---

e) Exercise 5.6.6, sections a, b

A country has two political parties, the Demonstrators and the Repudiators. Suppose that the national senate consists of 100 members, 44 of which are Demonstrators and 56 of which are Repudiators.

(a)

How many ways are there to select a committee of 10 senate members with the same number of Demonstrators and Repudiators?

$\binom{44}{5} * \binom{56}{5}$

(b)

Suppose that each party must select a speaker and a vice speaker. How many ways are there for the two speakers and two vice speakers to be selected?

$\binom{44}{1} * \binom{55}{1} * \binom{56}{1} * \binom{43}{1}$

Question 6:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.7.2, sections a, b

A 5-card hand is drawn from a deck of standard playing cards.

(a)

How many 5-card hands have at least one club?

$$\binom{52}{5} * \binom{39}{5}$$

(b)

How many 5-card hands have at least two cards with the same rank?

$$\binom{52}{5} - \binom{13}{5} * 4^5$$

b) Exercise 5.8.4, sections a, b

20 different comic books will be distributed to five kids.

(a)

How many ways are there to distribute the comic books if there are no restrictions on how many go to each kid (other than the fact that all 20 will be given out)?

$$5^{20}$$

(b)

How many ways are there to distribute the comic books if they are divided evenly so that 4 go to each kid?

$$\frac{20!}{4!4!4!4!4!}$$

Question 7:

How many one-to-one functions are there from a set with five elements to sets with the following number of elements?

a) 4

The domain has 5 elements, it is one more than the target(co domain), which only 4 here.
It won’t make sure that all 5 elements will have their distinctive element to map with. So this won’t be a one-to-one function.

---

b) 5
1st element has 5 path
2st has 4
3rs has 3
4th has 2
5th has 1.
So, 5!=120

---

c) 6

6*5*4*3*2=720

---

d) 7

7*6*5*4*4=2520

Appendix A.

The expected output of the call `printYearCalender(2016, 5)` is:

```
January 2016
Mon Tue Wed Thr Fri Sat Sun
         1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

February 2016
Mon Tue Wed Thr Fri Sat Sun
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29

March 2016
Mon Tue Wed Thr Fri Sat Sun
     1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

April 2016
Mon Tue Wed Thr Fri Sat Sun
                            1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

May 2016
Mon Tue Wed Thr Fri Sat Sun
                                1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
June 2016
```

Mon Tue Wed Thr Fri Sat Sun 1 2 3
4 5
6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28
29 30

July 2016
Mon Tue Wed Thr Fri Sat Sun 1 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27
28 29 30 31

August 2016
Mon Tue Wed Thr Fri Sat Sun 1 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27
28 29 30 31

September 2016
Mon Tue Wed Thr Fri Sat Sun 1 2 3
4
5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28
29 30

October 2016
Mon Tue Wed Thr Fri Sat Sun 1 2
3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31
November 2016
Mon Tue Wed Thr Fri Sat Sun 1 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27
28 29 30

December 2016
Mon Tue Wed Thr Fri Sat Sun 1 2 3
4
5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28
29 30 31