

Exam #2

Thursday, March 25, 2021

- This exam has 6 questions, with 100 points total.
- **You should submit your answers in the Gradescope platform (not on NYU Classes).**
- 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 for section (a), and the answer to section (b) typed as a comment.
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:
 - CLion or Visual-Studio. You should create a new project and work **ONLY** in it.
 - Three sheets of scratch paper.
 - Scientific CalculatorBesides 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 $\text{theta}(n)$, 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 (14 points)

Let a_n be the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, ...).

$$\text{That is: } a_n = \begin{cases} 1 & (n = 1 \vee n = 2) \\ a_{n-1} + a_{n-2} & n > 2 \end{cases}$$

Use mathematical induction to show that for every positive integer n : $a_n \geq (\sqrt{2})^n$

Question 2 (12 points)

You are creating a 4-digit pin code. How many choices are in the following cases?

Explain your answers.

- No digit is repeated.
- No digit is repeated, and they must appear in an increasing order.

Question 3 (15 points)

In the following game, 2 fair dice are tossed until either they both show an even number, or until you tried 4 times.

Let X be the random variable that denotes the **number of tosses** made in the game.

- Find the distribution of X . That is, for each possible value of X , say what is the probability X would get that value.
- What is $E(X)$? That is, find the expected value of X .

Explain your answers.

Question 4 (14 points)

a) Analyze the running time of `func`.

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 func(int n){
    int i, j, count;

    count = 0;
    for (i = 1; i <= log(n); i *= 2)
        for (j = 1; j <= i; j++)
            count += 1;

    return count;
}
```

b) Ben-El was asked to write a program that would initialize an array with the first 10 squares [1, 4, 9, 16, 25, 36, 49, 64, 81, 100], and then print these values.

He wanted to break this task to functions, and came with the following code:

```
int main(){
    int* arr;

    arr = first10Squares();
    printArray(arr, 10);

    return 0;
}

int* first10Squares(){
    int arr[10];

    for(int i = 0; i < 10; i++) {
        arr[i] = (i+1)*(i+1);
    }

    return arr;
}

void printArray(int arr[], int arrSize){
    for(int i = 0; i < arrSize; i++)
        cout<<arr[i]<<" ";
    cout<<endl;
}
```

When he ran the code, he was very surprised to find that the output was:

1 32766 3644 1 7964 1 -1709152400 32767 7964 1

1. Explain why the program did not print the expected output.
2. How could Ben-El fix this problem, keeping the breakdown to functions as he suggested? Write only the code that needs to be changed.

Note: you are not allowed to change the signature (header) of any function.

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 (30 points)

In this question, you should write a program that will read point-grade information (an integer in the range 0-100) of some students taking a course. It then finds the students with the highest grade.

The input would be entered as a non-empty sequence of lines, where each line would contain two numbers, separated by a space:

- A student ID (a positive integer)
- Their course point grade (an integer in the range 0-100)

To indicate the end of the input, the user should enter -1 as a student ID (the first number in the line indicating the end of the input).

After reading the input, the program would print the highest grade, followed by a list of all the student IDs of those who got that highest grade.

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

```
Please enter a non-empty sequence of lines.  
Each line should have a student ID and their grade (0-100),  
separated by a space.  
To indicate the end of the input, enter -1 as a student ID:  
7288 93  
62802 83  
90113874 83  
17539804 72  
163349801 93  
264791 85  
-1  
The highest grade is 93.  
The students with grade 93 are: 7288 163349801.
```

Notes:

1. Make sure to best **design your program**. In particular, break your implementation to functions.
2. Your program has to run in **worst-case linear time**. That is, if there are n lines in the input, the runtime should be $\Theta(n)$.
3. You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.

Question 6 (15 points)

Consider the following definitions:

- 1) Two elements in an array are considered **opposite-pair**, if they are at the same distance from the beginning, and from the end.

That is:

- The first and last elements are an opposite-pair
- The second and the second-last elements are an opposite pair
- The third and the third-last elements are an opposite pair
- etc.

- 2) The **palindrome distance** of a sequence of numbers, is the numbers of opposite-pairs, in which their two elements are not equal.

For example:

- The palindrome distance of [1, 3, 6, 6, 3, 1] is **0**, since $1=1$, $3=3$ and $6=6$, there are 0 opposite pair in which their elements are not equal.
- The palindrome distance of [1, 3, 6, 6, 5, 2] is **2**, since $1 \neq 2$, $3 \neq 5$ and $6=6$, there are 2 opposite pair in which their elements are not equal.

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

```
int palindromeDistance(int* arr, int n)
```

The function is given arr, a base address of an array containing integers, and its logical size n.

When calling palindromeDistance, it should return the palindrome distance of the sequence that is in arr.

For example, if `arr = [1, 3, 6, 6, 5, 2]`, the call `palindromeDistance(arr, 6)` should return 2 (the two opposite-pairs are (1, 2) and (3, 5)).

Implementation requirements:

- For simplicity, you may assume that arr has even number of elements (n is even).
- Your function should run in **worst case linear time**. That is, it should run in $\theta(n)$.
- Your function **must be recursive**.

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