

CS101 Practice Problems-set II

1 Iteration

1. Find 2^{nd} largest number from an array of general size n . (You have done this on a fixed size array with just an 'if then else' branching tree.)
2. Write a program to find out the frequency of occurrence of a given value v inside an array of size N . Read N , the actual numbers, and value v from the keyboard before you compute the frequency.
3. Modify the above program to compute the frequencies of all distinct numbers that appear in the array.
4. Print a triangle of '*'s of height 'r' rows. Now modify your program to print it upside down. of given size 'r', where r represents the no. of rows in the triangle.
5. Write a program to display Pascal's triangle. Pascal's triangle represents the binomial coefficients. The first few rows of Pascal's triangle are displayed below. Observe how a row is related to the row above it.

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
```

6. How many arrays did you use for the above problem? Try solving the problem with just one array if you have used more.
7. Given an array of integers of size n , find out if the numbers in the array appear in a palindromic order. A palindrome is a sequence that reads the same when you flip it. For example, 121 is a palindrome, 3 is a palindrome, and 234432 is also a palindrome.

8. Given two sorted arrays of sizes m and n , write a program that merges the two into another array of size $m + n$ such that this new array also remains sorted.

2 Functions

1. Develop functions to compute factorial n , and the Fibonacci series till n terms using iteration.
2. Convert the program that computes second largest given 5 numbers into a function. Test the function by using it in a separate file.
3. Convert the series $(e/\pi/\sqrt{x})$ that you wrote earlier into functions. All series converted into functions need to be in one single file called 'series.cpp'. Test all of them in one file.

3 Recursion

1. Solve the tower of Hanoi problem discussed in the class. How many moves do you have to make to solve a problem of n disks?
2. Compute Fibonacci(n) given n . How many calls are required for obtaining this n^{th} number in the series?
3. Compute the maximum value from an array of n integers.
4. Compute the average of the numbers in an array of size n . No global variables!
5. Flip an array. Do not use an additional array. Also attempt an iterative version for the same.
6. Sort an array. Count the number of comparisons required in your solution.
7. Compute gcd of 2 numbers by Euclid's method discussed in the class. First develop a recursive solution, then attempt an iterative version of the same.

4 Errors and Debugging

Following bugs were noticed in some programs. Can you make a guess about the reasons for these errors. Which part of the program will you concentrate on to fix these symptoms?

1. The program dealt with finding maximum value from an array. The value printed finally wasn't from the array.
2. The program dealt with an array. A segmentation fault was reported.
3. In main the statement `cout << fib(n) << count << endl` always printed `count` to be zero independent of the value of `n`. After separating printing of `count` into another statement placed later, the problem was solved.
4. It was a do while iteration, the program compiled, but when executed, it didn't terminate. The iteration wasn't an obvious infinite loop.
5. The program worked for some cases, but it didn't for some other.