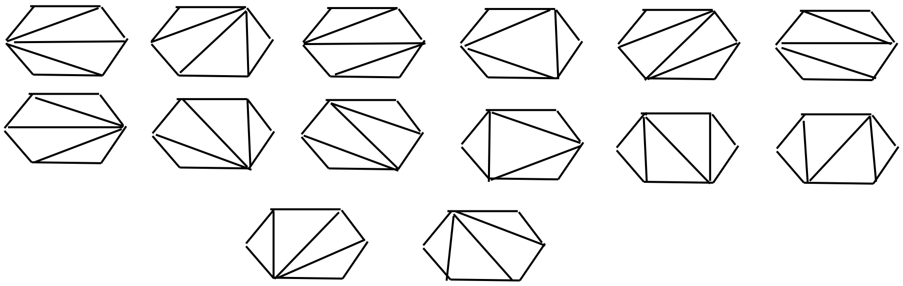
Please consider/**do not consider** Siska's grades in my final grade

CPSC 335, Homework No.1 Name Nicholas Ayson

1. (10 points) A polygon with n+2 sides can be divided into n triangles. Let C0=1, C1=1, C2=2, C3=5, draw all possible shapes for n=4 to find C4 Catalan number



2. (10 points) Number of ways to form a “mountain range” with n “/” and “\”

C0 =1	C1=1 /\
C2 =2 /\ /\ /\ / \	C3=5 /\ /\ /\ /\/\ / \ /\/\/\ / \/\ /\/ \ / \ / \

Draw all possible mountains for C4 Catalan number

```
                                                /\                                      /\
               /\        /\         /\/\       /    \      /\         /\  /\     /\/\      /    \      /\/\/\
/\/\/\/\, /\/\/   \,   /\/    \/\, /\/      \,  /\/        \, /    \/\/\, /   \/   \, /     \/\, /       \/\, /       \,


                                    /\
      /\       /\         /\/\       /    \
   /\/   \    /   \/\     /    \    /      \
/       \, /       \, /       \, /         \
```

3. (10 points) Given an unsorted file of size 105 records. Compute the number of phases to decide whether we should use PFD of order 3 or order 4

Order 3

1 0 0	1
1 1 1	3
2 2 1	5

4 3 2	9
7 6 4	17
13 11 7	31
24 20 13	57
44 37 24	105

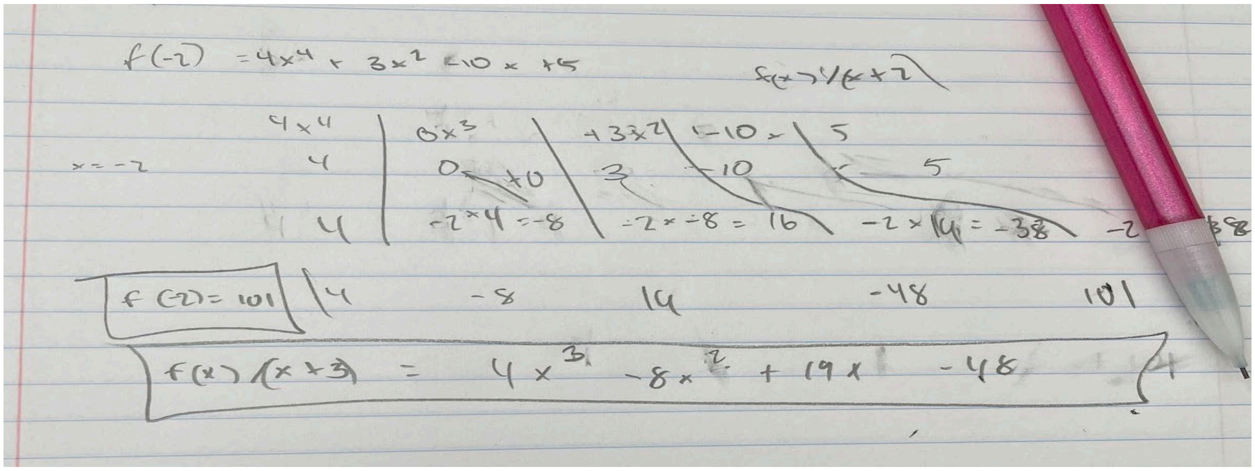
Order 4

1 0 0 0	1
1 1 1 1	4
2 2 2 1	7
4 4 3 2	13
8 7 6 4	25
15 14 12 8	49
29 27 23 15	94
56 52 44 29	181

We would use order 3

4. (10 points) Find at least one application of (a) Ugly numbers, (b) happy numbers. Please include your resources.
   A. Ugly numbers are the same as Hamming Numbers used for building an infinite ascending sequence of all 5 smooth numbers with Dijkstra.
   B. Happy Numbers are used for theoretical mathematics brought to attention by Reg Allenby

5. (10 points) Find the value of f(-2) where f(x) = $4X^4 + 3X^2 – 10 X +5$ by using Horner’s algorithm. Show the quotient and the remainder of f(x)/(x+2)
   f (-2) = 101
   f(x)/(x+2) = 4x^3-8x^2+19x-48



$f(-2) = 4x^4 + 3x^2 - 10x + 5$ $\quad$ $f(x)/(x+2)$

$4x^4 \mid 0x^3 \quad +3x^2 \quad -10x \quad 5$

$x = -2 \quad 4 \mid 0 \quad +0 \quad 3 \quad -10 \quad 5$

$4 \mid -2 \times 4 = -8 \quad -2 \times -8 = 16 \quad -2 \times 14 = -38 \quad -2 \quad 38$

$\boxed{f(-2) = 101} \quad 4 \quad -8 \quad 14 \quad -48 \quad 101$

$\boxed{f(x)/(x+3) = 4x^3 - 8x^2 + 19x - 48}$

6. Find the run-time of the following

Part a	Part b	Part c
$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n/2) + 1 & \end{cases}$ O(logn) T(n) = [T(n/2^2) +1] + 1 T(n) = T(n/2^2) + 2 T(n) = T(n/2^3) + 3 … T(n) = T(n/2^k) + k T(n) = 1 + logn	$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n/2) + n & \end{cases}$ O(n) T(n/2) + n [T(n/2^2) + n/2] + n T(n/2^3) + n/2^2 + n/2+n T(n/2^k) + n/2^k-1 + n/2^k+1 + …+n/2+ n T(1) = n[1/2^k-1 + 1/2^k-2 + …+1/2+1] T(n) = 1+n[1 + 1] T(n) = 1 + 2n O(n)	$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 4T(n/2) + n^2 & \end{cases}$ T(n) = 4T(n/2) + n2 = n2 + 4[4T(n/4) + $n^2/4$] = 2n2 + 16T(n/4) = ... = k·n2 + 4kT(n/2k) = ... k = log2n T(n) = O(n^2logn)

Programming part

7. (10 points) Write a program to determine whether int n=19 is a happy number or not. Try your program for n=4 and n=5555. Display new numbers at each step

   Sample output

```
Input: n = 19
```

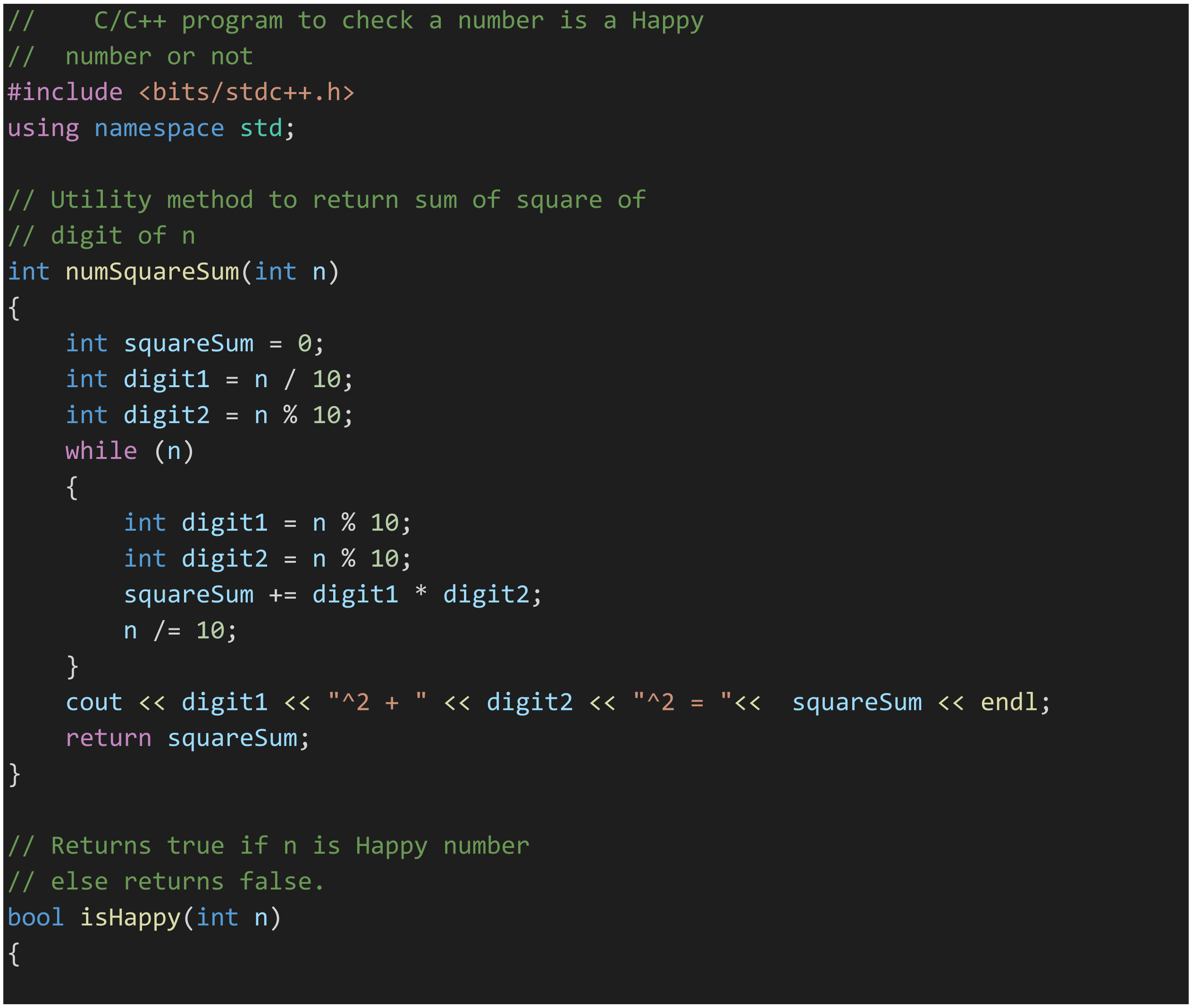
1^2+ 9^ 2= 82

8^2 + 2^2 = 68

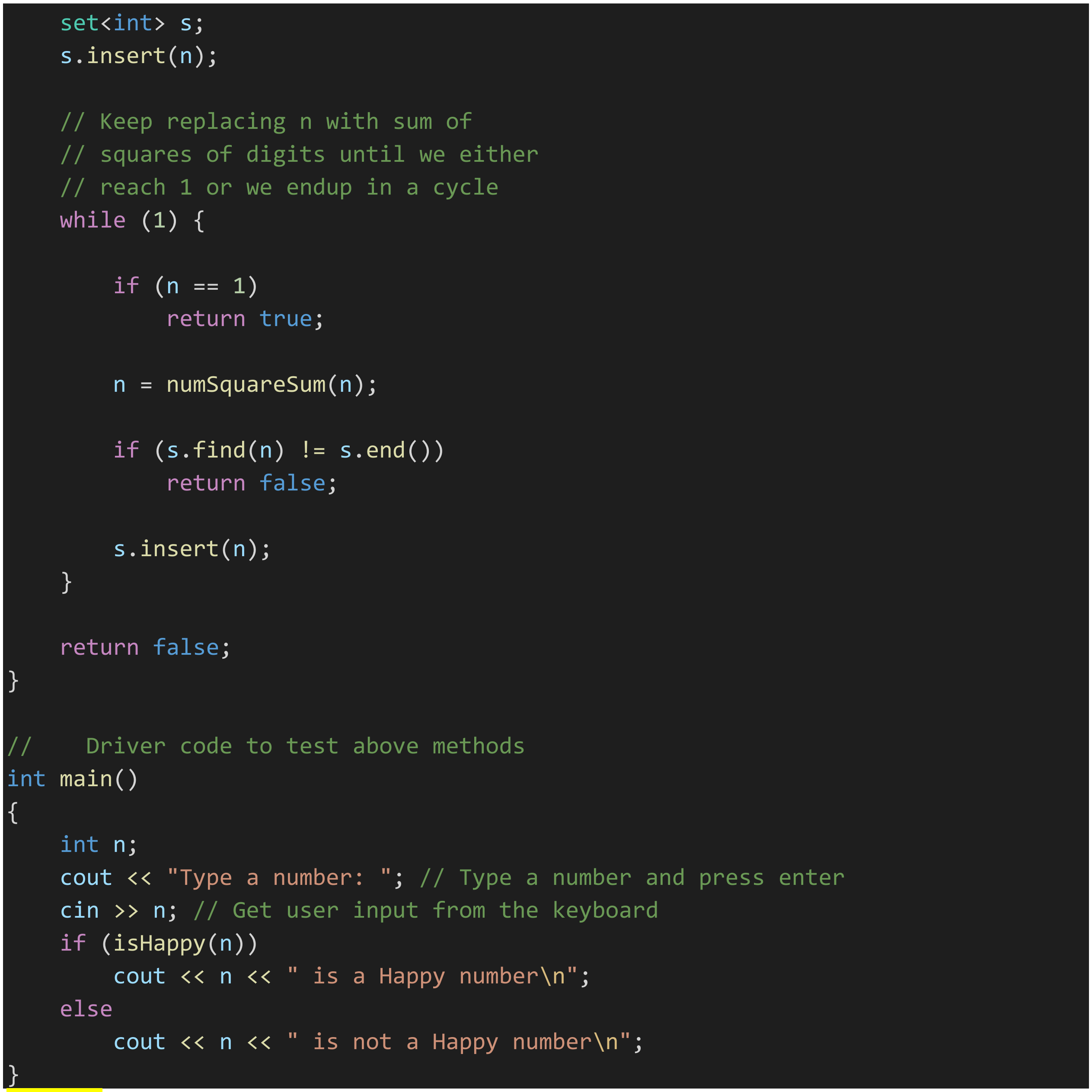
6^2 + 8^2 = 100

1^^{2}2 + 0^2+ 0^2 = 1

19 is a happy number



```
//    C/C++ program to check a number is a Happy
//  number or not
#include <bits/stdc++.h>
using namespace std;

// Utility method to return sum of square of
// digit of n
int numSquareSum(int n)
{
    int squareSum = 0;
    int digit1 = n / 10;
    int digit2 = n % 10;
    while (n)
    {
        int digit1 = n % 10;
        int digit2 = n % 10;
        squareSum += digit1 * digit2;
        n /= 10;
    }
    cout << digit1 << "^2 + " << digit2 << "^2 = "<<  squareSum << endl;
    return squareSum;
}

// Returns true if n is Happy number
// else returns false.
bool isHappy(int n)
{
```



```
    set<int> s;
    s.insert(n);

    // Keep replacing n with sum of
    // squares of digits until we either
    // reach 1 or we endup in a cycle
    while (1) {

        if (n == 1)
            return true;

        n = numSquareSum(n);

        if (s.find(n) != s.end())
            return false;

        s.insert(n);
    }

    return false;
}

//    Driver code to test above methods
int main()
{
    int n;
    cout << "Type a number: "; // Type a number and press enter
    cin >> n; // Get user input from the keyboard
    if (isHappy(n))
        cout << n << " is a Happy number\n";
    else
        cout << n << " is not a Happy number\n";
}
```

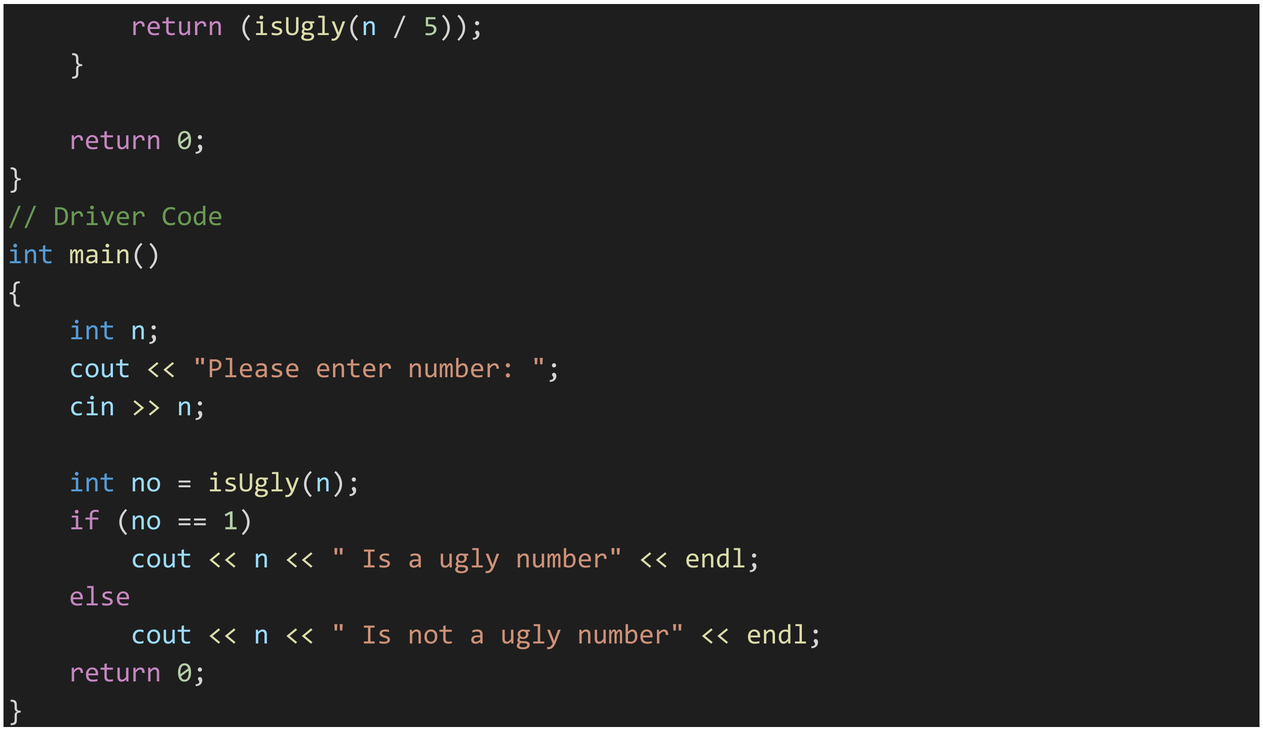
OUTPUT:

```
PS C:\Users\Nick\Desktop\Happy Number> g++ Happy.cpp -o Happy
PS C:\Users\Nick\Desktop\Happy Number> ./Happy
Type a number: 19
1^2 + 9^2 = 82
8^2 + 2^2 = 68
6^2 + 8^2 = 100
10^2 + 0^2 = 1
19 is a Happy number

PS C:\Users\Nick\Desktop\Happy Number> ./Happy
Type a number: 4
0^2 + 4^2 = 16
```

```
1^2 + 6^2 = 37
3^2 + 7^2 = 58
5^2 + 8^2 = 89
8^2 + 9^2 = 145
14^2 + 5^2 = 42
4^2 + 2^2 = 20
2^2 + 0^2 = 4
4 is not a Happy number

PS C:\Users\Nick\Desktop\Happy Number> ./Happy
Type a number: 5555
555^2 + 5^2 = 100
10^2 + 0^2 = 1
5555 is a Happy number
```

8. (10 points) Write a program to determine whether a given number is an ugly number or not
   Try your program for n= 13, n=24, n=5832

```
// C++ implementation to check
// if a number is an ugly
// number or not

#include <stdio.h>
#include <stdlib.h>
#include <bits/stdc++.h>
using namespace std;


int isUgly(int n)
{
    // Base Cases
    if (n == 1)
        return 1;
    if (n <= 0)
        return 0;

    //has to be divided by 2 3 or 5
    if (n % 2 == 0) {
        return (isUgly(n / 2));
    }
    if (n % 3 == 0) {
        return (isUgly(n / 3));
    }
    if (n % 5 == 0) {
```



```
        return (isUgly(n / 5));
    }

    return 0;
}
// Driver Code
int main()
{
    int n;
    cout << "Please enter number: ";
    cin >> n;

    int no = isUgly(n);
    if (no == 1)
        cout << n << " Is a ugly number" << endl;
    else
        cout << n << " Is not a ugly number" << endl;
    return 0;
}
```

OUTPUT:

```
PS C:\Users\Nick\Desktop\Ugly Numbers> g++ ugly.cpp -o ugly
PS C:\Users\Nick\Desktop\Ugly Numbers> ./ugly
Please enter number: 13
13 Is not a ugly number

PS C:\Users\Nick\Desktop\Ugly Numbers> ./ugly
Please enter number: 24
24 Is a ugly number

PS C:\Users\Nick\Desktop\Ugly Numbers> ./ugly
Please enter number: 5832
5832 Is a ugly number
```

9. The run-time of recursive function to find the nth Fibonacci number is O(2^n). The non-recursive function's run-time is O(n). Now there exist a O(log n) algorithm for computing the nth Fibonacci number. It is based on the equality

$$\begin{bmatrix} F(n-1) & F(n) \\ F(n) & F(n+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n \quad \text{, for n=1 F(0)=0, F(1)=1, F(2)=1}$$

Use the above equation to find the first 8 Fibonacci number by hand (1 1 2 3 5 8 15 23).

*****LAST TWO NUMBERS OF YOUR EXAMPLE ARE WRONG**

Should be 13 and 21***

$f(n) = f(n-1) + f(n-2)$

$0, \ldots\ldots, (n-1)$

$$f(n) = \begin{array}{cc} f(3) & f(2) \\ \left(\begin{matrix} 1 \\ 1 \end{matrix}\right. & \left.\begin{matrix} 1 \\ 0 \end{matrix}\right)^{(n-1)} \\ f(2) & f(1) \end{array} \qquad \begin{pmatrix} f(n+1) & f_n \\ f_n & f_{n-1} \end{pmatrix}$$

$$2 = f(3) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 = \begin{pmatrix} \boxed{\textcircled{1}\ 1} \\ \textcircled{1\ 0} \end{pmatrix} \begin{pmatrix} \textcircled{1} & \boxed{1} \\ \textcircled{1} & \boxed{0} \end{pmatrix} = \begin{matrix} 1+1 & 1+0 \\ 1+0 & 1+0 \end{matrix} = \begin{pmatrix} \textcircled{2} & 1 \\ 1 & 1 \end{pmatrix}$$

$$3 = f(4) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^3 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^1 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \textcircled{3} & 2 \\ 2 & 1 \end{pmatrix}$$

$$5 = f(5) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^4 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^3 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \textcircled{5} & 3 \\ 3 & 2 \end{pmatrix}$$

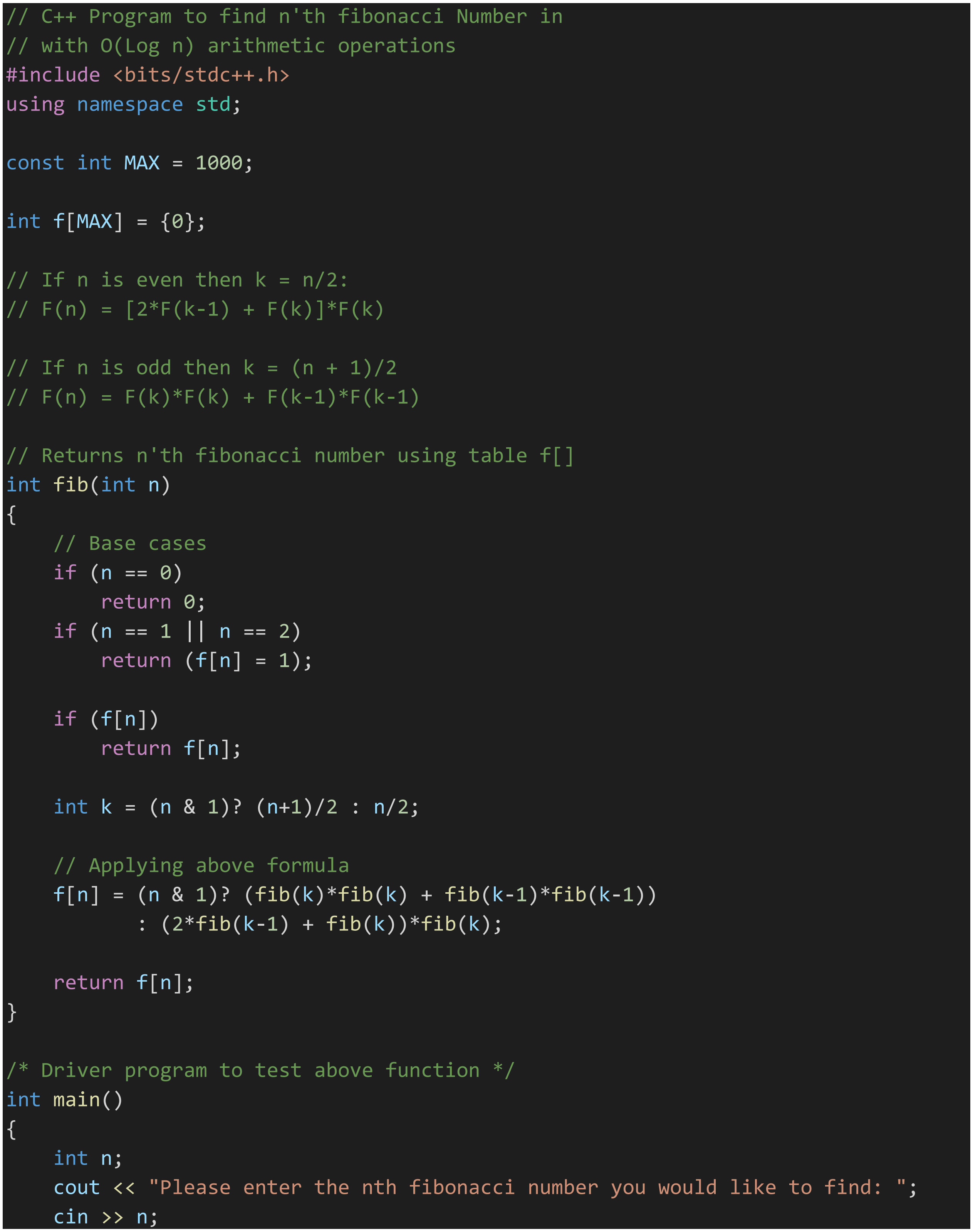
$$8 = f(6) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^5 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^4 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 3 \\ 8 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \textcircled{8} & 5 \\ 5 & 3 \end{pmatrix}$$

$$13 = f(7) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^6 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^5 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 5 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \textcircled{13} & 8 \\ 8 & 5 \end{pmatrix}$$

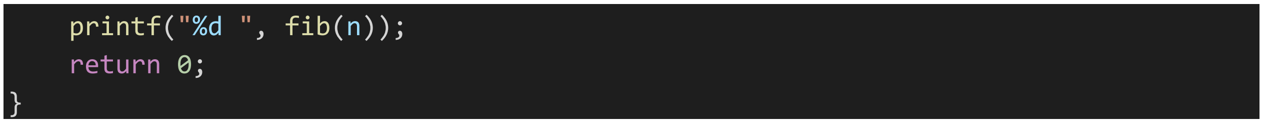
$$21 = f(8) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^7 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^6 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 13 & 8 \\ 8 & 5 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} \textcircled{21} & 13 \\ 13 & 8 \end{pmatrix}$$

Write a program to find the n=42 Fibonacci number (first test your program to make sure the first 8 Fibonacci numbers match with the above list)



```
// C++ Program to find n'th fibonacci Number in
// with O(Log n) arithmetic operations
#include <bits/stdc++.h>
using namespace std;

const int MAX = 1000;

int f[MAX] = {0};

// If n is even then k = n/2:
// F(n) = [2*F(k-1) + F(k)]*F(k)

// If n is odd then k = (n + 1)/2
// F(n) = F(k)*F(k) + F(k-1)*F(k-1)

// Returns n'th fibonacci number using table f[]
int fib(int n)
{
    // Base cases
    if (n == 0)
        return 0;
    if (n == 1 || n == 2)
        return (f[n] = 1);

    if (f[n])
        return f[n];

    int k = (n & 1)? (n+1)/2 : n/2;

    // Applying above formula
    f[n] = (n & 1)? (fib(k)*fib(k) + fib(k-1)*fib(k-1))
           : (2*fib(k-1) + fib(k))*fib(k);

    return f[n];
}

/* Driver program to test above function */
int main()
{
    int n;
    cout << "Please enter the nth fibonacci number you would like to find: ";
    cin >> n;
```



```
    printf("%d ", fib(n));
    return 0;
}
```

OUTPUT:

```
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 1
1
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 2
1
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 3
2
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 4
3
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 5
5
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 6
8
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 7
13
PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 8
21

PS C:\Users\Nick\Desktop\Fibonnacci number O(logn)> ./fib
Please enter the nth fibonacci number you would like to find: 42
267914296
```

Please save your program codes and their sample run to the same document as the rest of the problems,, save all in file CPSC335_H1_lastName and email to mahmadnia@fullerton.edu

By 6:00 PM on Monday before class. Thanks