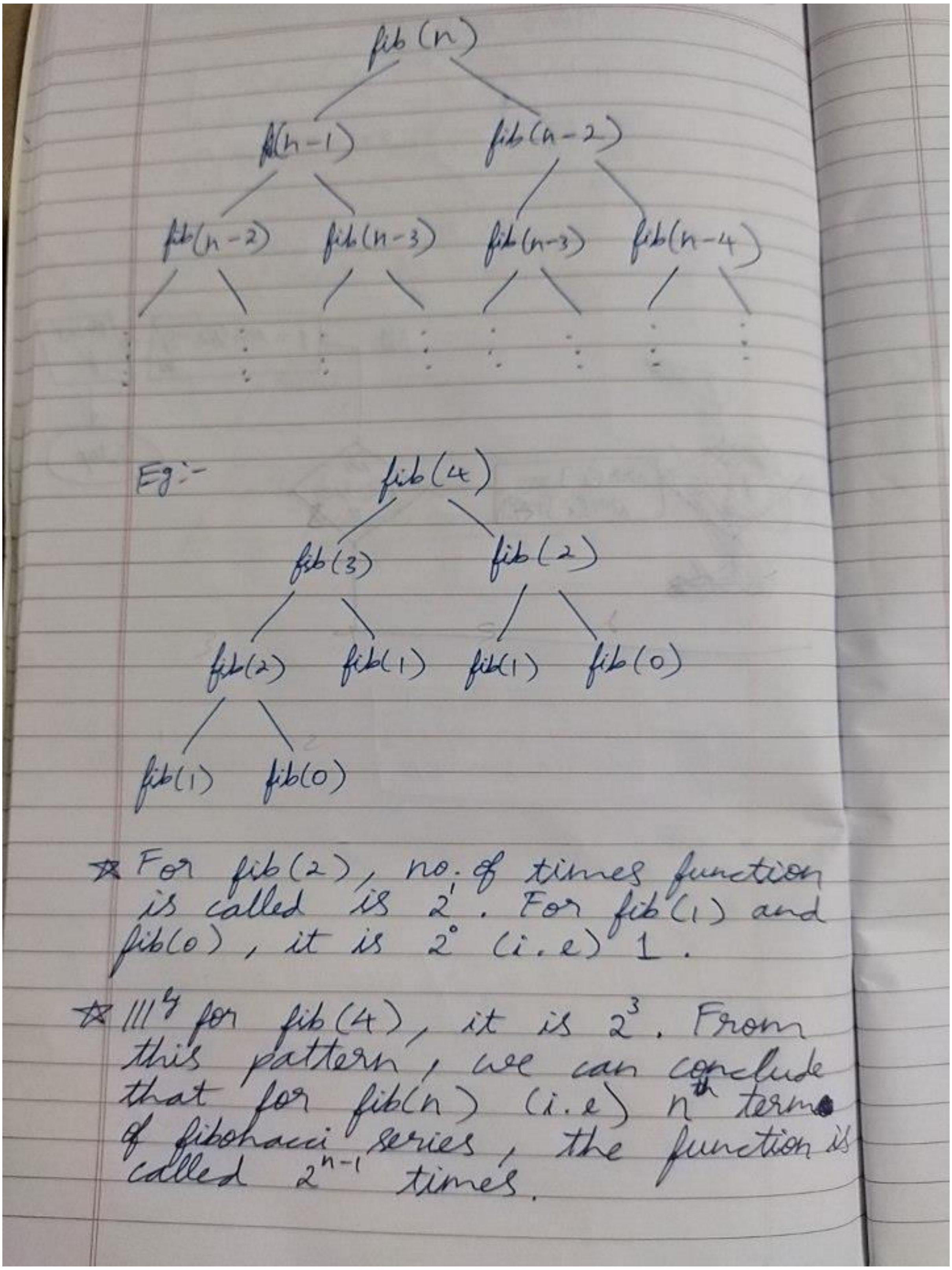
LAB ASSIGNMENT 12

1. Complex Numbers: Write a function that takes 2 complex numbers as input and prints the sum and multiplication of the 2 complex numbers.

```
#include <stdio.h>
#include <stdlib.h>
struct complex
{
  int real, img;
};

void sum_mul(struct complex a, struct complex b){
    struct complex c;

      c.real = a.real + b.real;
      c.img = a.img + b.img;

      if (c.img >= 0){
        printf("\nSum of the complex numbers = %d + %di", c.real, c.img);}
      else{
        printf("\nSum of the complex numbers = %d %di", c.real, c.img);}
      c.real = a.real*b.real + (-1*a.img*b.img);
      c.img = a.img*b.real + a.real*b.img;

      if (c.img >= 0){
        printf("\nMultiplication of the complex numbers = %d + %di",
c.real, c.img);}
      else{
        printf("\nMultiplication of the complex numbers = %d %di",
c.real, c.img);}

}

int main()
{
  struct complex a, b;



      printf("Enter a and b where a + ib is the first complex number.");

      printf("\na = ");
      scanf("%d", &a.real);

      printf("b = ");
```

```
    scanf("%d", &a.img);


    printf("Enter c and d where c + id is the second complex number.");


    printf("\nc = ");
    scanf("%d", &b.real);


    printf("d = ");
    scanf("%d", &b.img);


  sum_mul(a, b);


 return 0;
}
```

2. Fibonacci Analyzation: Compute Fibonacci(n) given n. How many calls are required for obtaining this nth number in the series? Draw a recurrence tree for the same.


fib(n)
fib(n-1)
fib(n-2)
fib(n-2)
fib(n-3)
fib(n-3)
fib(n-4)
Eg:-
fib(4)
fib(3)
fib(2)
fib(2)
fib(1)
fib(1)
fib(0)
fib(1)
fib(0)

★ For fib(2), no. of times function is called is 2^1. For fib(1) and fib(0), it is 2^0 (i.e) 1.

★ $|||^{ly}$ for fib(4), it is 2^3. From this pattern, we can conclude that for fib(n) (i.e) n^{th} term of fibonacci series, the function is called 2^{n-1} times.

```
#include <stdio.h>

int fibonacci(int i) {
  if(i == 0) {
    return 0;
  }

  if(i == 1) {
    return 1;
  }
  return fibonacci(i-1) + fibonacci(i-2);
}

int  main() {

  int i, n;

        printf("Enter number of terms: ");
        scanf("%d", &n);

  printf("The Fibonacci series upto %d terms:\n",n);
  for (i = 0; i < n; i++) {
    printf("%d\n", fibonacci(i));
  }
  return 0;
}
```

Samuela Abigail
71762108039

3. 3-Way Merge: Given three sorted arrays of sizes m, n and o, write a function that merges the three into another array of size m + n +o such that this new array also remains sorted.

```
#include<stdio.h>
void mergeArray(int a[], int n1, int b[], int n2, int c[], int n3, int mer[])
{
        int i=0, j=0, k=0, l=0, m=0,n4, temp;
        n4 = n1 + n2+ n3;


        k=0;
        for(i=0;i<n1;i++)     //Array Initialized
    {
      mer[k]=a[i];
      k++;
    }


    l=n1;
    for(i=0;i<n2;i++)     //Array Initialized
    {
       mer[l]=b[i];
       l++;
    }


    m=n1+n2;
```

```
for(int i=0;i<n3;i++)      //Array Initialized
{
    mer[m]=c[i];
    m++;
}

//mer[n4]=mer[m];
printf("\nThe merged array is\n");
for(int i=0;i<n4;i++){
printf("%d ",mer[i]);
}

printf("\nAfter sorting the sorted array is\n");
for(int i=0;i<n4;i++)        //sorts in descending order
{
    int temp;
    for(int j=i+1; j<n4 ;j++)
    {
        if(mer[i]<mer[j])
        {
            temp=mer[i];
            mer[i]=mer[j];
            mer[j]=temp;
        }
```

```
      }
   }

   for(int i=0 ; i<n4 ; i++)
   {
      printf(" %d ",mer[i]);
   }
}

int main()
{
        int n1, n2, n3, n4, i;

        printf("\nEnter size of First Array  :  ");
        scanf("%d", &n1);

   int a[n1];
        printf("\nEnter the elements for First Array :  ");
        for(i = 0; i < n1; i++)
        {
        scanf("%d", &a[i]);
        }

        printf("\nEnter size of second Array :  ");
```

```
    scanf("%d", &n2);

int b[n2];
    printf("\nEnter the elements for Second Array :  ");
    for(i = 0; i < n2; i++)
    {
    scanf("%d", &b[i]);
    }

    printf("\nEnter size of third Array :  ");
    scanf("%d", &n3);

int c[n3];
    printf("\nEnter the elements for third Array :  ");
    for(i = 0; i < n3; i++)
    {
    scanf("%d", &c[i]);
    }



    n4 = n1 + n2+ n3;
    int mer[n4];
    mergeArray(a, n1, b, n2, c, n3, mer);     //Function Call
```

```
        return 0;
}
```

4. Rearrangement: Given an array A[1:n] which contains a set of two characters (‘B’ and ‘G’) representing Boys and Girls standing in a row in no particular order. Devise a function that rearranges boys after all the girls in the row. Remember, your code should perform single-scan on that entire array!

```
#include <stdio.h>
#include <ctype.h>

int main()
{
    char name[] = {'B','G','G','B','G','G'};
    int size=0, i=0, count=0;
    size = sizeof(name);
    while(name[i] != '\0')
    {
        i++;
        if(name[i]=='G')
        ++count;
    }
    printf("count of girls = %d, size=%d \n", count, size);
    for (i=0; i<count; i++)
        name[i]='G';
    for (i=count; i<size; i++)
        name[i]='B';

    i=0;
    while(i<size)
    {
        printf("%c", name[i]);
        i++;
    }
    return 0;
}
```