DFS Lab

August 2, 2017

1. Solutions
   - Pascal's Triangle
   - Sorting
   - Prime Number
   - Fibonacci Numbers

```
#include <stdio.h>
int main()
{
    int rows, coef = 1, space, i, j;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=0; i<rows; i++)
    {
        for(space=1; space <= rows-i; space++)
            printf("  ");
        for(j=0; j <= i; j++)
        {
            if (j==0 || i==0)
                coef = 1;
            else
                coef = coef*(i-j+1)/j;

            printf("%4d", coef);
        }
        printf("\n");
    }
printf("\n\n");
    return 0;
}
```

```
#include <stdio.h>
int main()
 {
    int *data,i,n,steps,temp;
    printf("Enter the number of elements to be sorted: ");
    scanf("%d",&n);
    data = (int *)malloc(n * sizeof(int));
    for(i=0;i<n;++i)
      {
       printf("%d. Enter element: ",i+1);
       scanf("%d",&data[i]);
    }
    for(steps=0;steps<n;++steps)
    for(i=steps+1;i<n;++i)
     {
         if(data[steps]>data[i])
         /* To sort in descending order, change > to <. */
          {
             temp=data[steps];
             data[steps]=data[i];
             data[i]=temp;
         }
    }
    printf("In ascending order: ");
    for(i=0;i<n;++i)
        printf("%d  ",data[i]);

    printf("\n\n");
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n, i, flag = 0;

    printf("Enter a positive integer: ");
    scanf("%d",&n);

    for(i=2; i<=sqrt(n); ++i)
    {
        // condition for nonprime number
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }

    if (flag==0)
        printf("%d is a prime number.",n);
    else
        printf("%d is not a prime number.",n);

    printf("\n\n");
    return 0;
}
```

```
#include <stdio.h>
int main(){
    int i, n, t0 = 0, t1 = 1, nextTerm;
    printf("\n");
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("Fibonacci Sequence: ");
    for (i = 1; i <= (n + 1); ++i)
    {
        printf("%d ", t0);
        nextTerm = t0 + t1;
        t0 = t1;
        t1 = nextTerm;
    }
    printf("\n\n");
    return 0;
}
```