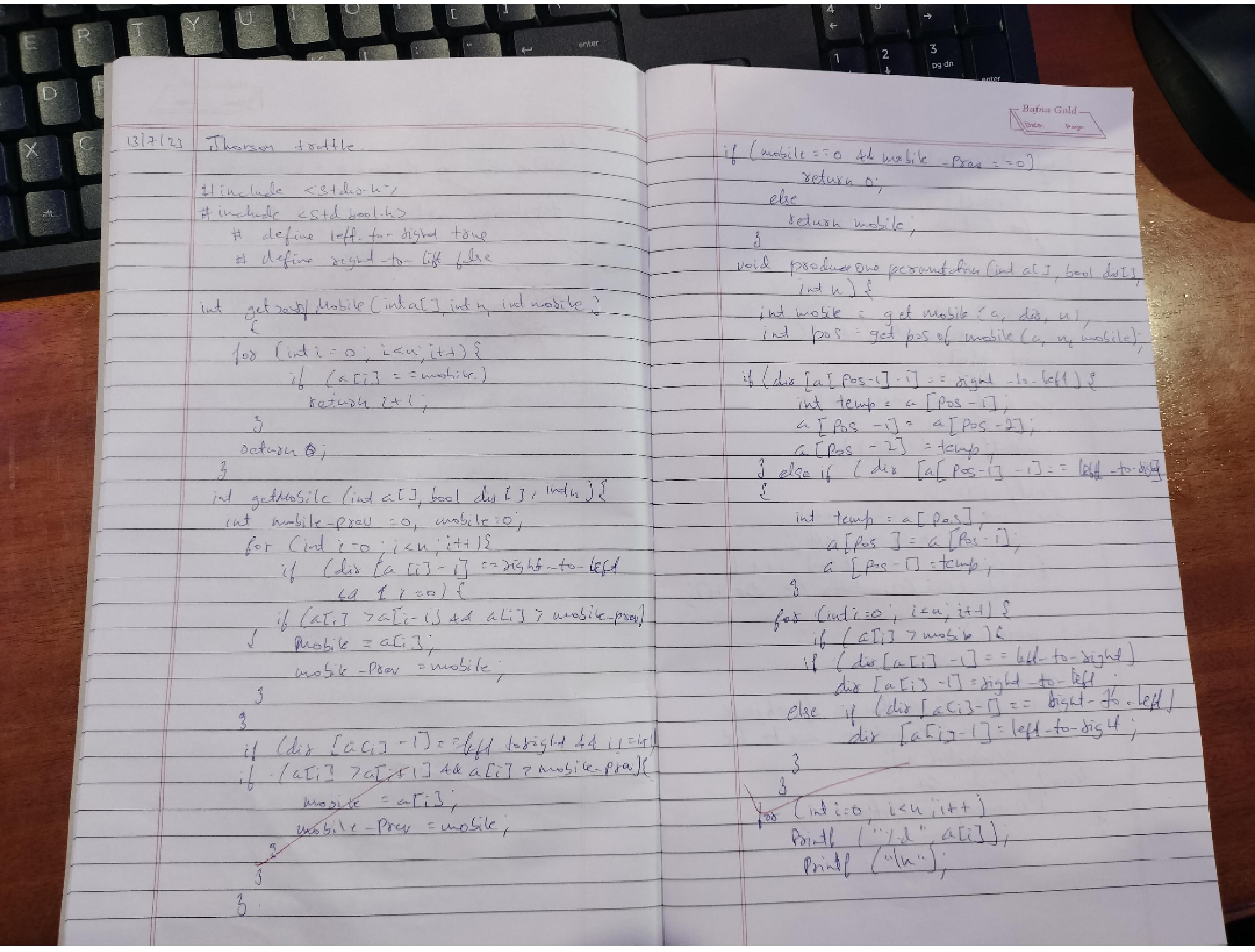


13/7/21 Jhonson trottle

```
#include <stdio.h>
#include <stdbool.h>
   # define left-to-right true
   # define right-to-left false

int getposOfMobile(int a[], int n, int mobile )
   {
   for (int i = 0; i<n; i++) {
        if (a[i] = =mobile)
          return i+1;
     }
   return 0;
  }
 int getMobile (int a[], bool dir [], int n ){
  int mobile-prev =0, mobile=0;
   for (int i=0; i<n; i++){
        if (dir [a [i] - 1] ==right-to-left
             && i !=0) {
       if (a[i] >a[i-1] && a[i] > mobile-prev)
    {    mobile = a[i];
         mobile-Prev =mobile;
     }
   }
   if (dir [a[i] -1] = =left toright && i!=n-1)
   if (a[i] >a[i+1] && a[i] > mobile-prev){
        mobile =a[i];
        mobile-Prev =mobile;
     }
    }
   }
if (mobile = =0 && mobile -Prev = =0)
        return 0;
   else
      return mobile;
  }
void produceOnepermutation (int a[], bool dir[],
        int n) {
  int mobile = get mobile (a, dir, n);
  int pos = get pos of mobile (a, n, mobile);

 if (dir [a[Pos-1] -1] = = right -to-left) {
      int temp = a [Pos - 1];
      a [Pos -1] = a [Pos -2];
      a [Pos - 2] =temp;
   } else if ( dir [a[ Pos-1] -1] = = left -to-right]
   {
      int temp = a[Pos];
         a[Pos ] = a [Pos -1];
         a [Pos -1] =temp;
     }
  for (int i=0; i<n; i++) {
       if (a[i] > mobile){
      if (dir[a[i] -1] = = left-to-right)
         dir [a[i] -1] =right -to-left ;
    else if (dir [a[i]-1] == right-to-left)
          dir [a[i]-1]=left-to-right;
     }
    }
  for (int i=0; i<n; i++)
    Printf ("%d", a[i]);
     Printf ("\n");
```

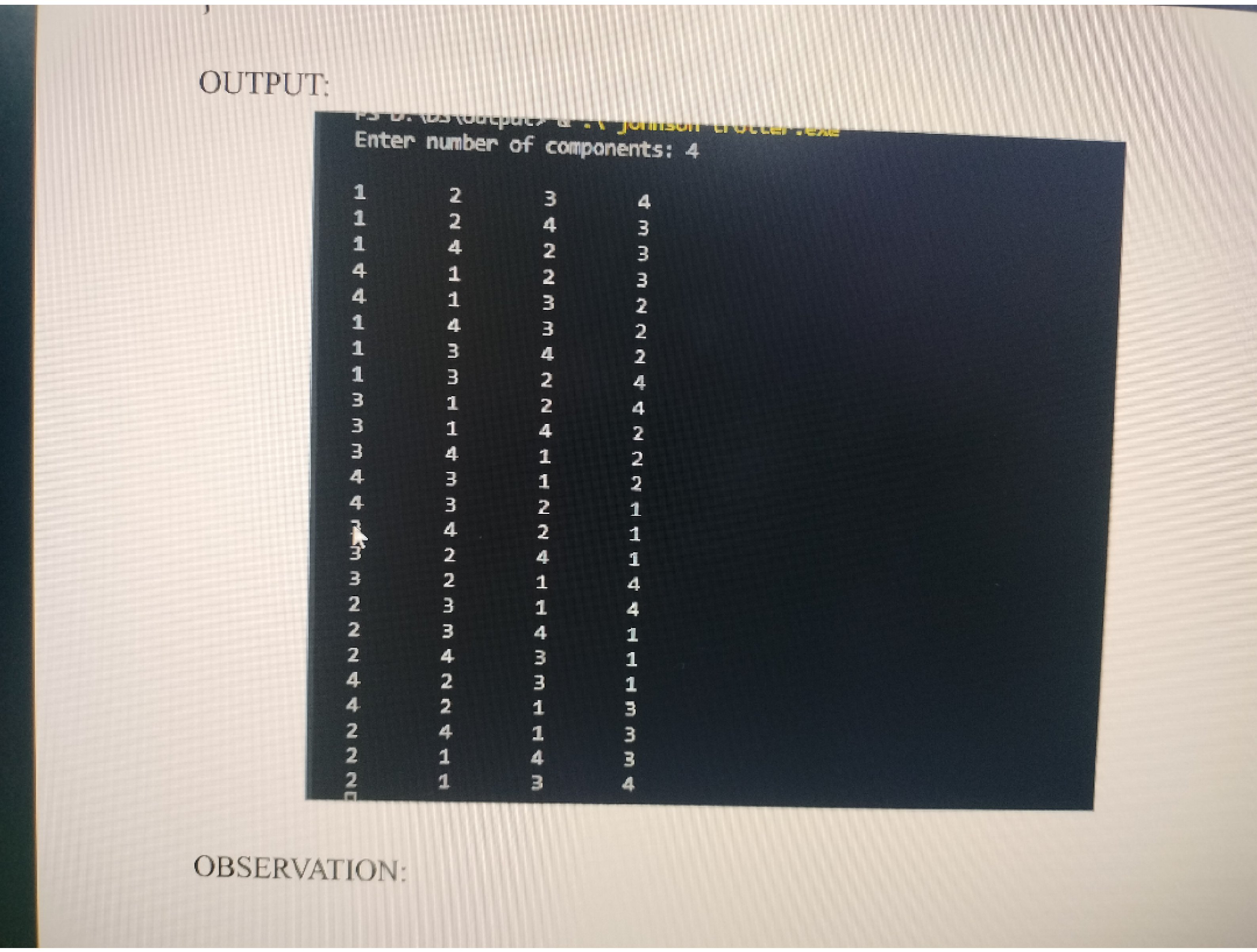
```
int fact (int n)
 {
   int result = 1;
    for (int i=1; i<n; i++)
  {
    result *= i;
   }
   return result;
void  produce permutation (int n) {
      int  a[n];
   bool  dir[n];
   for  (int i=0; i<n; i++) {
      a[i]  = i+1;
      Printf ("%d", a[i]);
   }
  Printf ("\n");
  for (int i =0; i<n ; i++)
      dir[i] = right-to-left;
   for (int i=1; i< fact (n) ; i++ ]
      Produce One Permutation (a, dir, n);

Void  main ()
 {
   int n;
  Printf ("\nEnter the no of objects
          whose Permutation are to be
          generated :");
 Scanf ("%d", &n);
     Produce permutation (n);
}
```

output

```
 1 2 3
 1 3 2
 3 1 2
 3 2 1
 2 3 1
 2 1 3
```

[illegible]
13/7/23



}

OUTPUT:

```
PS D:\DS\Output> & .\johnson trotter.exe
Enter number of components: 4

1        2        3        4
1        2        4        3
1        4        2        3
4        1        2        3
4        1        3        2
1        4        3        2
1        3        4        2
1        3        2        4
3        1        2        4
3        1        4        2
3        4        1        2
4        3        1        2
4        3        2        1
3        4        2        1
3        2        4        1
3        2        1        4
2        3        1        4
2        3        4        1
2        4        3        1
4        2        3        1
4        2        1        3
2        4        1        3
2        1        4        3
2        1        3        4
```

OBSERVATION: