Algorithm 1

Divij Singh

31/01/18

1 Polynomials

1.1

def poly(num):
$P_n(x) = poly(num - 1) + a_n * x^n$

1.2

def poly(num):
$P_n(x) = poly(num - 1) + a_n * x^{n-1} * x$

1.3

1.
def poly(num,a,n,x):
num[x]=poly(num,(a),(n-1),(x-1)) + $a[n] * (x * *n)$
return num[x]

2.
def poly(num,a,n,x):
num[x]=poly(num,(a),(n-1),(x-1)) + $a[n] * (x * *(n-1)) * x$
return num[x]

3.
def poly(n,x):
for i in n
p=$(p * x)$+i
return p

2 one-mapping

2.1

for i in list array.length do
if list array[i] is not in mapping array, remove list array[i] and mapping array[i]
end for

2.2

```
def one_one_map(list_array ,map_array:
        i=0
        while i< len(list_array):
                is_there=False

                j=0
                while j<len(list_array):
                        if list_array[i]==mapping_array[j]:
                                is_there=True

                        j++

                if is_there!=True:
                        del list_array[i]
                        del mapping_array[i]

                        one_one_map(list_array ,mapping_array)
                i++

return list_array
```

(list array has the elements of the left side, while mapping array has the corresponding mappings.)