Roll no: BCSF18M005

Name: Humail Shahzad

Quiz title: QUIZ 2

CHALLENGE-A

Part A

Time complexity: $O(n^3)$

Space complexity: O(n)

Part B

1)

```
int l;
 {
 if ( i == 0)
 {
 l=3 * ( j * col + k);
 }
 else if ( i == 1)
 {
 l = 3 * (j * col + k) + 1;
 }
 else
 l= 3 * (j * col + k) + 2;
```

2)

(K IS THE BYTE NUMBER AND COL IS THE NUMBER OF COLUMNS GIVEN)

(BYTES ARE STARTING FROM INDEX 0)

```
int i , j, k;
 if ((K + 3) % 3 == 1)
 {
 i = 1;
 j = K / (3 * col);
 k = (K - 1) / 3 - (j * col);

}
 if ((K + 3) % 3 == 0)
 {
 i = 0;
 j = K / (3 * col);
 k = K / 3 - (j * col);
 }
```

```
if ((K + 3) % 3 == 2)
{
i = 2;
j = K / (3 * col);
k = (K - 2) / 3 - (j * col);
}
```

Part C

True

False

True

False

CHALLENGE-B

K=degree

	TIME	memory
int calPower(int base, int exponent)		
{		
int res=1;	1	4
for (int i = 0; i < exponent; i++)	k+1	4
{		
res = res * base;	k	
}		
return res;	1	
}		
double evaluatePolynomial(double coefficients[], int exponents[], int N, int x)		
{		
int m=pow(x, exponents[0]);	2k+3	12
double res=0;	1	8
for (int k = 0; k < N; k++)	n+1	4
{		
if ((k - 1) >= 0)	1*n	
{		
int l = exponents[k - 1] - exponents[k];	1*n	4
for (int j = 0; j < l; j++)	n*(k)	4
{		
m = m / x;	n*k	
}		
}		
res = res + coefficients[k] * m;	n	
}		
return res;	1	
}		
	2K+2NK+4N+6	
	(K=N)	

$O(N^2)$ O(1)

CHALLENGE-D

VERSION 1(USING USER DEFINED STRUCTURE)

```
include<iostream>
using namespace std;
#include"queue.h"
void displayPrimes(int n)
{
 Queue<int> a;
 for (int i = 2; i <= n; i++)
 {
 a.enQueue(i);
 }
 Queue<int> prime;
 int m = 2;
 while (m < sqrt(n))
 {
 Queue<int> temp;
 m = a.getElementAtFront();
 a.deQueue();
 prime.enQueue(m);
 while (!a.isEmpty())
 {
 if (!(a.getElementAtFront() % m == 0))
 {
 temp.enQueue(a.getElementAtFront());
 }
 a.deQueue();
 }
 while (!temp.isEmpty())
 {
 a.enQueue(temp.getElementAtFront());
 temp.deQueue();
 }
 }
 while (!prime.isEmpty())
 {
 cout << prime.getElementAtFront()<<" ";
 prime.deQueue();
 }
```

```
 while (!a.isEmpty())
 {
 cout << a.getElementAtFront()<<"  ";
 a.deQueue();
 }
}
int main ()
{
 int n;
 cin >> n;
displayPrimes(n);
 return 0;
}
```

VERSION 2

```
#include<iostream>
using namespace std;
#include<queue>
void displayPrimes(int n)
{
 queue<int>a;
 for (int i = 2; i <= n; i++)
 {
 a.push(i);
 }
 queue<int>prime;
 int m = 2;
 while (m < sqrt(n))
 {
 queue <int >temp;
 m = a.front();
 a.pop();
 prime.push(m);
 while (!a.empty())
 {
 if (!(a.front() % m == 0))
 {
 temp.push(a.front());
 }
 a.pop();
 }
 while (!temp.empty())
 {
 a.push(temp.front());
 temp.pop();
```

```
 }
 }
 while (!prime.empty())
 {
 cout << prime.front()<<" ";
 prime.pop();
 }
 while (!a.empty())
 {
 cout << a.front()<<"  ";
 a.pop();
 }
}
int main()
{
 int n;
 cin >> n;
displayPrimes(n);
 return 0;
}
```