Name: MD RAKIBUL ISLAM

ID: 20183290424

Data structure Homework

1. Create a list and Implemented by array.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a[1000];

        int i;

     int* pa = a;

     for(i = 0; i <= 1000; i++)
     {
          if(i==500){
              continue;
          }
          printf("a[%d] \n",i);
          pa++;
     }
}
```

Result:

Output will be

```
a[479]
a[480]
a[481]
a[482]
a[483]
a[484]
a[485]
a[486]
a[487]
a[488]
a[489]
a[490]
a[491]
a[492]
a[493]
a[494]
a[495]
a[496]
a[497]
a[498]
a[499]
a[501]
a[502]
a[503]
a[504]
a[505]
a[506]
a[507]
a[508]
a[509]
a[510]
a[511]
a[512]
a[513]
a[514]
a[515]
a[516]
a[517]
a[518]
a[519]
a[520]
a[521]
a[522]
a[523]
```

Insertion 500 in position

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a[1000];

        int i, x, pos;

     int* pa = a;

     for(i = 0; i <= 1000; i++)
     {
         if(i==500){
             continue;
         }
         printf("a[%d] \n",i);
         pa++;
     x = 500; // element to be inserted
     pos = 500;      // position at which element is to be inserted
     for (i = 1000; i >= pos; i--)     // shift elements forward
         a[i] = a[i - 1];
     a[pos - 1] = x;      // insert x at pos
     for (i = 0; i <= 1000; i++)      // print the updated array
         printf("%d ", a[i]);
     printf("\n");

     return 0;
     }
}
```

Time complexity of array is:
O(n).(worst time)