

Name: MD RAKIBUL ISLAM

ID: 20183290424

## Data structure Homework -2

1. Create a list (0, 1, ..., 499, 501, ..., 999, 1000).  
Implemented by linked lists.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  typedef struct node Node;
4  struct node {
5      int data;
6      Node* next;
7  };
8  Node *head;
9  //head = 0;
10 Node *create_node(int item, Node *next)
11 {
12     for(int item=0; item<500; item++){
13         Node *new_node = (Node *)malloc(sizeof(Node));
14         if (new_node == NULL){ //if there are any error new_node will assign NULL
15             printf("Error! Could Not Create A New Node\n");
16             exit(1);
17         }else{
18             new_node->data = item;
19             new_node->next = next;
20             printf("data = %d\n", new_node->data);
21         }
22     }
23     for(int item=501; item<1000; item++){
24         Node *new_node = (Node *)malloc(sizeof(Node));
25         if (new_node == NULL){ //if there are any error new_node will assign NULL
26             printf("Error! Could Not Create A New Node\n");
27             exit(1);
28         }else{
29             new_node->data = item;
30             new_node->next = next;
31             printf("data = %d\n", new_node->data);
32         }
33     }
34     return 0;
35 }
36
37 int main()
38 {
39     create_node(0, 1);
40     return 0;
41 }
```

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

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a[1000];
7
8      int i, x, pos;
9
10     int* pa = a;
11
12     for(i = 0; i <= 1000; i++)
13     {
14         if(i==500){
15             continue;
16         }
17         printf("a[%d] \n",i);
18         pa++;
19     }
20     x = 500; // element to be inserted
21     pos = 500; // position at which element is to be inserted
22     for (i = 1000; i >= pos; i--) // shift elements forward
23         a[i] = a[i - 1];
24     a[pos - 1] = x; // insert x at pos
25     for (i = 0; i <= 1000; i++) // print the updated array
26         printf("%d ", a[i]);
27     printf("\n");
28
29     return 0;
30 }
```

Time complexity of this array is:  $O(n)$ .