Traverse

```c
#include <stdio.h>
int main () {
    int arr[] = {1, 2, 3, 4, 5};
    printf ("The elements of the array is : ");
    for (int i = 0; i < 5; i++) {
        printf ("%d ", arr[i]);
    }
    return 0;
}
```

Insert

```c
#include <stdio.h>
int main () {
    int arr[6] = {1, 2, 3, 4, 5};
    int pos = 2;
    int value = 77;

    int idx = 5 - 1;
    while (idx >= pos) {
        arr[idx + 1] = arr[idx];
        idx--;
    }

    arr[pos] = value;
    printf ("The elements of the array after inserting : ");
    for (int i = 0; i < 5 + 1; i++) {
        printf ("%d ", arr[i]);
    }
    return 0;
}
```

Delete

```c
#include <stdio.h>
int main () {
    int arr[] = {1, 2, 3, 4, 5};
```

```
    int pos = 2;

    for (int i = pos; i < 5 - 1; i++) {
        arr[i] = arr[i + 1];
    }

    printf ("The elements of the array after deleting : ");
    for (int i = 0; i < 5 - 1; i++) {
        printf ("%d ", arr[i]);
    }
    return 0;
}
```

Search

```
#include <stdio.h>

int main () {
    int arr[] = {1, 2, 3, 4 , 5};
    int value = 4;
    int loc = -1;
    for (int i = 0; i < 5; i++) {
        if (arr[i] == value) {
            loc = i + 1;
            break;
        }
    }
    if (loc == -1) {
        printf ("Value not found");
    } else {
        printf ("Value found at location %d", loc);
    }
    return 0;
}
```

Fibonacci

```c
#include <stdio.h>
int fibonacci(int n) {
    if (n <= 1) return n;

    return fibonacci(n - 1) + fibonacci(n - 2);;
}

int main () {
    int n;
    printf ("Enter a number : ");
    scanf ("%d", &n);
    int ans = fibonacci (n);
    printf ("The %dth element of fibonacci sequence is %d", n, ans);
    return 0;
}
```

Factorial

```c
#include <stdio.h>
int factorial (int n) {
    if (n <= 1) return 1;

    return n * factorial (n - 1);
}
int main () {
    int n;
    printf ("Enter a number : ");
    scanf ("%d", &n);
    int ans = factorial (n);
    printf ("The factorial of %d is %d", n, ans);
    return 0;
}
```