```rust
fn main() {
    let arr:[i32;4]=[10,20,30,40];
    println!("array is {:?}",arr);
    println!("array size is :{}",arr.len());
}
```

```
array is [10, 20, 30, 40]
array size is :4
```

2.

```rust
fn main() {
    let arr:[i32;4]=[10,20,30,40];
    println!("array is {:?}",arr);
    println!("array size is :{}",arr.len());
    for index in 0..4{
        println!("index is :{} &value is {}",index,arr[index]);
    }
}
```

```
array is [10, 20, 30, 40]
array size is :4
index is :0 &value is 10
index is :1 &value is 20
index is :2 &value is 30
index is :3 &value is 40
```

```rust
fn main() {
    let mut arr=[10,20,30];
    update(& mut arr);
    println!("inside main{
        :?}",arr);
    }
    fn update(arr:&mut[i32;3]
        ){
        for i in 0..3{
            arr[i]=0;
        }
        println!("inside
            update {:?}",arr
            );
}
```

```
inside update [0, 0, 0]
inside main[0, 0, 0]
```

```rust
use std::io;

fn main()
{
    let mut numbers: Vec<i32> = get_array_input();
    println!("Unmodified array: {:?}", numbers);
    numbers = numbers.iter_mut()
        .map(|n| if *n == 2 { *n * 3 } else { *n })
        .collect();
    numbers.push(5);
    println!("Modified array: {:?}", numbers);
}

fn get_array_input() -> Vec<i32>
{
    let mut input = String::new();
    println!("Enter numbers separated by spaces:");
    io::stdin().read_line(&mut input).unwrap();
    input.trim().split_whitespace().map(|s| s.parse().unwrap()).collect()
}
```

```
Enter numbers separated by spaces:
2 4 6 8
Unmodified array: [2, 4, 6, 8]
Modified array: [6, 4, 6, 8, 5]
```

5.

```rust
use std::io;

fn input_array() -> [i32; 5] {
    let mut arr = [0; 5];
    for i in 0..arr.len() {
        println!("Enter element {}: ", i + 1);
        let mut input = String::new();
        io::stdin().read_line(&mut input).expect("Failed to read
            line");
        arr[i] = input.trim().parse().expect("Please enter a
            valid number");
    }
    arr
}

fn print_array(arr: &[i32; 5]) {
    println!("Array elements are:");
    for &element in arr {
        println!("{}", element);
    }
```

```rust
fn main() {
    let array = input_array();
    print_array(&array);
}
```

Output:

```
Enter element 1:
10
Enter element 2:
20
Enter element 3:
30
Enter element 4:
40
Enter element 5:
50
Array elements are:
10
20
30
40
50
```

6.

```rust
use std::io;

fn input_array() -> [i32; 5] {
    let mut arr = [0; 5]; // Define an array of size 10 with
        initial values 0
    for i in 0..arr.len() {
        println!("Enter element {}: ", i + 1);
        let mut input = String::new();
        io::stdin().read_line(&mut input).expect("Failed to read
            line");
        arr[i] = input.trim().parse().expect("Please enter a
            valid number");
    }
    arr
}

fn modify_array(arr: &mut [i32; 5]) {
    println!("Modifying array elements...");
    for element in arr.iter_mut() {
```

```rust
fn modify_array(arr: &mut [i32; 5]) {
    println!("Modifying array elements...");
    for element in arr.iter_mut() {
        *element *= 2;
    }
}

fn print_array(arr: &[i32; 5]) {
    println!("Array elements are:");
    for &element in arr {
        println!("{}", element);
    }
}

fn sum_array(arr: &[i32; 5]) -> i32 {
    let mut sum = 0;
    for &element in arr {
        sum += element;
    }
    sum
```

```rust
fn main() {
    let mut array = input_array();
    modify_array(&mut array);
    print_array(&array);

    let sum = sum_array(&array);
    println!("Sum of array elements: {}", sum);
}
```

```
Enter element 1:
10
Enter element 2:
20
Enter element 3:
30
4Enter element 4:
40
Enter element 5:
50
Modifying array elements...
Array elements are:
20
40
60
80
100
Sum of array elements: 300
```