Problem1

```
public static boolean checkModify(int n)
{
	System.out.println("int methods");
	if (n > 0)
		return false;
	else
	{
		n = -n;
		return true;
	}
}
```

Problem2

```
public static double evalPol(double A[],int n, double x)
{
	double pol = A[0];
	int d = 1;

	for (int i = 1 ; i <= n ; i++)
	{
		pol += A[i] * Math.pow(x,d);
		d++;
	}

	return pol;
}
```

Problem3

```
	public static <T> void exchange(T a[],int pos1,int pos2)
	{
		T temp;
		temp = a[pos1];
		a[pos1] = a[pos2];
		a[pos2] = temp;
	}

	public static <T> int counting(T a[], T target)
	{
		int count = 0;

		for (int i = 0 ; i < a.length ; i++)
		{
			if (a[i].equals(target))
				count++;
		}

		return count;
	}

	public static <T> void reverse(T[] A, int n)
	{
		try
		{
			int i = 0;
			int j = n;
			T temp;

			while(i < j)
			{
				temp = A[i];
				A[i] = A[j];
				A[j] = temp;

				i++;
				j--;
			}
		}
		catch(ArrayIndexOutOfBoundsException e)
		{
			System.out.println("Value of n is out of range in the
array");
		}
	}
```

Problem4

```
public class GArray<T>
{
	private T array[];

	public GArray(int size)
	{
		try
		{
			array = (T[]) new Object[size];
		}
		catch(NegativeArraySizeException e)
		{
			array = (T[]) new Object[50];
		}
	}

	public T get(int i)
	{
		try
		{
			return array[i];
		}
		catch(IndexOutOfBoundsException e)
		{
			System.out.println("Wrong index");
			return null;
		}
	}

	public void set(int i , T e)
	{
		try
		{
			array[i] = e;
		}
		catch(IndexOutOfBoundsException exp)
		{
			System.out.println("Can set new value");
		}
	}
}
```

Problem5

```
public class Pair<T,U>
{
     public T first ;
     public U second ;

     public Pair ( T first , U second )
     {
          this.first = first;
          this.second = second;
     }

     public static <T,U> GArray<Pair<T,U>> pair(T[] A, U[] B, int n)
     {
          try
          {
               GArray<Pair<T, U>> g = new GArray<>(n);

               for (int i = 0 ; i < n ; i++)
                    g.set(i, new Pair<T,U>(A[i], B[i]));

               return g;
          }
          catch(IndexOutOfBoundsException e)
          {
               System.out.println("Index out of range in the array");
               return null;
          }
     }
}
```

Problem6

```
public class Problem6
{
	public static void test()
	{
		String ss[] = {"A”,"K","B","M","W","J"};
		sort(ss, ss.length, false);
	}

	public static <T extends Comparable<T>> void sort(T[] A, int n,
boolean incr)
	{
		T temp;

		for (int i = 0 ; i < n - 1 ; i++)
		{
			for (int j = i+1 ; j < n ; j++)
			{
				if (incr)
				{
					if (A[i].compareTo(A[j]) > 0)
					{
						temp = A[i];
						A[i] = A[j];
						A[j] = temp;
					}
				}
				else
				{
					if (A[i].compareTo(A[j]) < 0)
					{
						temp = A[i];
						A[i] = A[j];
						A[j] = temp;
					}
				}
			}
		}
	}
}
```