```
package Fall2015Final;

public class RentalContract
{
	private int id;
	private String carType;
	private int numOfDays;

	public RentalContract()
	{
		this(-1,null,-1);
	}

	public RentalContract(int id, String carType, int numOfDays)
	{
		setId(id);
		setCarType(carType);
		setNumOfDays(numOfDays);
	}

	public boolean equals(RentalContract r)
	{
		if (this.id == r.id)
			return true;
		else
			return false;
	}

	public void printContract()
	{
		System.out.println("Rental contract");
		System.out.println("ID             : " + id);
		System.out.println("Car type       : " + carType);
		System.out.println("Number of days : " + numOfDays);
	}

	public int getId()
	{
		return id;
	}

	public void setId(int id)
	{
		this.id = id;
	}

	public String getCarType()
	{
		return carType;
	}

	public void setCarType(String carType)
	{
		this.carType = carType;
	}

	public int getNumOfDays()
	{
		return numOfDays;
```

```
	}

	public void setNumOfDays(int numOfDays)
	{
		this.numOfDays = numOfDays;
	}
}
```

```
package Fall2015Final;

public class CarRentalCompany
{
	private RentalContract contracts[];
	private int nContracts;
	private static final int MAX_SIZE = 100;

	public CarRentalCompany()
	{
		contracts = new RentalContract[MAX_SIZE];
		nContracts = 0;
	}

	public void rentCar(int contractId,int numOfDays, String carType)
	{
		if (nContracts < contracts.length)
		{
			int index = findContract(contractId);

			if (index != -1) //found
				System.out.println(" ERROR: there is another contract with
the same id. Use extend contract instead");
			else
			{
				RentalContract r = new RentalContract(contractId,carType,
numOfDays);
				contracts[nContracts] = r;
				//or
				//contracts[nContracts] = new
RentalContracts(contractId,carType, numOfDays);

				nContracts++;
			}
		}
		else
			System.out.println("ERROR, array is full");
	}

	public void returnCar(int contractId)
	{
		int index = findContract(contractId);

		if (index != -1)
		{

		}
	}
```

```
public void printReceipt(RentalContract rentalContract)
{
	rentalContract.printContract();

	double cost;

	if (rentalContract.getNumOfDays() < 30)
		cost = 100 * rentalContract.getNumOfDays();
	else
	{
		int n = rentalContract.getNumOfDays() / 30;
		int d = rentalContract.getNumOfDays() % 30;
		cost = n * 2000 + d * 100;
	}

	System.out.println("Cost : " + cost);
}

private int findContract(int contractId)
{
	for (int i = 0 ; i < nContracts ; i++)
	{
		if (contracts[i].getId() == contractId)
			return i;
	}

	return -1;
}

public void extendContract(int contractId, int numOfDays)
{
	int index = findContract(contractId);

	if (index != -1)
	{
		int n = contracts[index].getNumOfDays() + numOfDays;

		contracts[index].setNumOfDays(n);
	}
}

public int findContract(RentalContract contract)
{
	for (int i = 0 ; i < nContracts ; i++)
	{
		if (contracts[i].equals(contract))
			return i;
	}

	return -1;
}

public void printContracts()
{
	System.out.println("List of all contracts");

	for (int i = 0 ; i < nContracts ; i++)
		contracts[i].printContract();
```

```
            System.out.println("----------------------");
      }

      public RentalContract[] getLongContracts()
      {
            RentalContract [] r = new RentalContract[nContracts];
            int j = 0;

            for (int i = 0 ; i < nContracts ; i++)
            {
                  if (contracts[i].getNumOfDays() >= 90)
                  {
                        r[j] = new
RentalContract(contracts[i].getId(),contracts[i].getCarType(),contracts[i].getNumOf
Days());
                        j++;
                  }
            }

            return r;
      }

      public static void main(String[] args)
      {
            CarRentalCompany cr = new CarRentalCompany();

            cr.rentCar(1000, 15, "Corolla");
            cr.rentCar(1001, 100, "Accent");

            RentalContract r[] = cr.getLongContracts();
            for (int i = 0 ; i < r.length ; i++)
            {
                  r[i].printContract();
            }

            cr.returnCar(1001);
            cr.printContracts();
      }
}
```

```
package Fall2015Final;

public class Q3
{

      public static void main(String[] args)
      {
            char word[] = {'l','e','v','e','l'};

            System.out.println(isPalindrome2(word));
      }

      public static boolean isPalindrome(char[] word)
      {
            int i = 0 , j = word.length - 1;

            while(i < j)
            {
```

```
                if (word[i] != word[j])
                        return false;

                i++;
                j--;
        }

        return true;
    }

    public static boolean isPalindrome2(char[] word)
    {
        int i = 0;

        while(i < word.length/2){
                if (word[i] != word[word.length - i - 1])
                        return false;

                i++;
        }

        return true;
    }
}
```