1.

(*Math: pentagonal numbers*) A pentagonal number is defined as $n(3n-1)/2$ for $n = 1, 2, \ldots$, and so on. Therefore, the first few numbers are 1, 5, 12, 22, ... Write a method with the following header that returns a pentagonal number:

```
public static int getPentagonalNumber(int n)
```

Write a test program that uses this method to display the first 100 pentagonal numbers with 10 numbers on each line.

2.

(*Palindrome integer*) Write the methods with the following headers

```
// Return the reversal of an integer, i.e., reverse(456) returns 654
public static int reverse(int number)

// Return true if number is a palindrome
public static boolean isPalindrome(int number)
```

Use the `reverse` method to implement `isPalindrome`. A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and reports whether the integer is a palindrome.

3. Napisati metodu koja ima sledeci header:
public static String removeChar(String str, char ch)
Metoda prima string i karakter i vraca novi string koji predstavlja originalni string sa izuzetkom proslijedjenog karaktera.
Napisati program koji pita korisnika da unese string i karakter, a zatim ispisuje string iz koga je obrisan uneseni karakter.
Sample run:
Unesi string:
stepenice Medicinske skole
Unesi karakter:
s
Rezultat: tepenice Medicinke kole

4.

(*Display characters*) Write a method that prints characters using the following header:

```
public static void printChars(char ch1, char ch2, int
 numberPerLine)
```

This method prints the characters between `ch1` and `ch2` with the specified numbers per line. Write a test program that prints ten characters per line from `1` to `Z`. Characters are separated by exactly one space.

5.

(*Phone keypads*) The international standard letter/number mapping for telephones is shown in Programming Exercise 4.15. Write a method that returns a number, given an uppercase letter, as follows:

```
int getNumber(char uppercaseLetter)
```

Write a test program that prompts the user to enter a phone number as a string. The input number may contain letters. The program translates a letter (uppercase or lowercase) to a digit and leaves all other characters intact. Here is a sample run of the program:

```
Enter a string: 1-800-Flowers Enter
1-800-3569377
```

```
Enter a string: 1800flowers Enter
18003569377
```

6.

(*Convert milliseconds to hours, minutes, and seconds*) Write a method that converts milliseconds to hours, minutes, and seconds using the following header:

```
public static String convertMillis(long millis)
```

The method returns a string as *hours*:*minutes*:*seconds*. For example, `convertMillis(5500)` returns a string 0:0:5, `convertMillis(100000)` returns a string `0:1:40`, and `convertMillis(555550000)` returns a string `154:19:10`.

7.

(*Emirp*) An *emirp* (prime spelled backward) is a nonpalindromic prime number whose reversal is also a prime. For example, 17 is a prime and 71 is a prime, so 17 and 71 are emirps. Write a program that displays the first 100 emirps. Display 10 numbers per line, separated by exactly one space, as follows:

```
13 17 31 37 71 73 79 97 107 113
149 157 167 179 199 311 337 347 359 389
...
```

8.

(*Palindromic prime*) A *palindromic prime* is a prime number and also palindromic. For example, 131 is a prime and also a palindromic prime, as are 313 and 757. Write a program that displays the first 100 palindromic prime numbers. Display 10 numbers per line, separated by exactly one space, as follows:

```
2 3 5 7 11 101 131 151 181 191
313 353 373 383 727 757 787 797 919 929
...
```

9. Napisati metodu koja prima jedan argument, broj pitanja, te generiše toliko nasumičnih, jednostavnih pitanja oduzimanja tipa : „Koliko je 5 - 2 ?“. Metoda treba da broji broj tačnih i netačnih odgovora te ih ispiše korisniku.

10. Napisati metodu koja prima jedan argument te simulira bacanje novčića toliko puta. Nakon što se simulacija završi, program ispisuje koliko puta je novčić pokazao glavu a koliko puta pismo.