*CSE 102*

**FUNCTIONS AND RECURSION**

1. Which statement should we put in place of the question marks (??) so that after the loop is executed the variable evenSum holds the sum of all even positive numbers less than 100?

**int** evenSum = 0;

**for**(**int** i=0; i<100; ??)

evenSum += i;

1. i+=2
2. i++
3. i--
4. ++i
5. Which statement should we put in place of the question marks (??) so that after the loop is executed the variable evenSum holds the sum of all even positive numbers less than 100?

**int** evenSum = 0;

**for**(**int** i=0; ??; i++)

evenSum += 2\*i;

1. i<100
2. i<=100
3. i<200
4. i<50
5. i<=50
6. What will be the output of the following program?

**public** **static** **void** main(String[] args){

String s = "";

**for**(**int** i=0; i<'z'-'a'+1; i++)

s += (**char**)('a'+i);

System.***out***.println(s);

}

1. Compiler error. Because - operator cannot be used between two chars.
2. Compiler error. Because + operator cannot be used between a char and int.
3. Lowercase English alphabet letters in a row.
4. Compiler error. Because int type cannot be casted to char.
5. What does the following function do?

**static** **int** func(String s, **char** c) {

**int** res = 0;

**for**(**int** i=0; i<s.length(); i++)

**if**(s.charAt(i) == c)

res++;

**return** res;

}

1. Counts the number of different characters the string *s* has.
2. Counts the total number of characters the string *s* has.
3. Counts the number of times the character *c* appears in the string *s*.
4. Counts the number of times a character other than *c* appears in the string *s*.
5. What does the following function do?

**static** String func(String s1, String s2, **int** n) {

String a = s1.substring(0, n);

String b = s1.substring(n, s1.length());

**return** a + s2 + b;

}

1. Appends string *s2* at the end of string *s1*.
2. Appends string *s1* at the end of string *s2*.
3. Inserts string *s1* between the *n*th and *(n+1)*th characters of string *s2*.
4. Inserts string *s2* between the *n*th and *(n+1)*th characters of string *s1*.
5. What will be the output of the following program?

**public** **static** **int** func(**int** n) {

**if**(n%2==0)

**return** 0;

**else** **if**(n%2==1)

**return** 1;

}

**public** **static** **void** main(String[] args){

System.***out***.println(*func*(32));

}

1. 0
2. 1
3. Compiler error because the operator == has higher precedence than %
4. Compiler error because compiler does not know math and it cannot see that the two conditionals are exhaustive for the input space. It is afraid of some input failing to satisfy both conditions and reaching the end of the function before reaching any return statement.

**BONUS:**

Prove that weird(n) returns 1 for all positive integers n.

**public** **static** **int** weird(**int** n) {

**if**(n==1)

**return** 1;

**else** **if**(n%2 == 0)

**return** *weird*(n/2);

**else**

**return** *weird*(n+1);

}

It is a famous [conjecture](https://en.wikipedia.org/wiki/Collatz_conjecture) in mathematics that the following function weirder(n) returns 1 for all positive integers n. No one has been able to prove it so far. Simple-looking recursive functions may exhibit complex behavior.

**public** **static** **int** weirder(**int** n) {

**if**(n==1)

**return** 1;

**else** **if**(n%2 == 0)

**return** *weirder*(n/2);

**else**

**return** *weirder*(3\*n+1);

}