Programing Languages 1
Exam 1

2015. autumn

I accept the rules and regulations of the university and the class. At the beginning of the semester I was informed about the criteria to pass (provided in written form on the homepage of the class).
Without signature this exam cannot be graded, and counted in to the final grade.

_____ _____

signature *name*

1. Write a C program that reads in an integer number from the keyboard. (1p) The number is a color code. (1 – red, 2 – green, 3 – blue, 4 – orange, 5 – purple, 6 – yellow, 7 – black, 8 – white). After reading the number print out the respective color. (2p)

   If the number is not listed above print out "No such color..." (1p)
   (IN THIS PROGRAM YOU **MUST NOT USE** THE "**IF**" STATEMENT)

   Example run:
   Give a number please... *4*
   Orange

2. Write a C program that reads in float numbers from the keyboard until -300. (1p) The numbers are degrees in Celsius denoting daily averages. (We suppose that -300 is not a valid daily average so this value can be used as a termination signal.). After the user enters -300 make the program print out how many of the averages were below 0°C (2p). Write your own logical function (int function returning 0 or 1) to decide whether the average is negative or not and use it in your program. (3p) 0°C is not negative.

   Example run:
   Give a value please... *-5.3*
   Give a value please... *15.2*
   Give a value please... *10.0*
   Give a value please... *7.2*
   Give a value please... *-0.5*
   Give a value please... *-300*
   Averages below 0 degree: 2

3. Write a C program that reads in 20 integer numbers (1p) and first prints out the previously read positive numbers (1p) and after the previously read negative numbers (1p). Use a function to decide whether a number is positive or negative (2p). When you read in the numbers print out exactly the same string that is shown below. Pay attention to start counting from 1. (2p)

   Example run:
   The 1. number is: *5*
   The 2. number is: *-5*
   The 3. number is: *13*
   The 4. number is: *7*
   The 5. number is: *-20*
   The 6. number is: *-11*
   ...
   The 19. number is: *9*
   The 20. number is: *-11*
   Positive numbers: 5 13 7 … 9
   Negative sum: -5 -20 -11 … -11

Programing Languages 1
Exam 1

2015. autumn

4. Create a C program that reads in 20 integer numbers to an array. (1p) After all the numbers are read print them back to the screen but only till the first 0 number. (2p) If there is no zeros between the numbers print back all of them. (1p) In the output print the numbers between parentheses separated by commas. Do not include 0. (see below). (2p)

Example run:

Give a number please... 5
Give a number please... -9
Give a number please... 11
Give a number please... 34
Give a number please... 2
Give a number please... -54
Give a number please... -23
Give a number please... 6
Give a number please... 10
Give a number please... -3
Give a number please... -4
Give a number please... -65
Give a number please... 0
Give a number please... 98
Give a number please... 3
Give a number please... -44
Give a number please... 76
Give a number please... 0
Give a number please... 98
Give a number please... 8
(5),(-9),(11),(34),(2),(-54),(-23),(6),(10),(-3),(-4),(-65)

5. Create a C program that reads in a string from the keyboard. The string only contain letters of the English alphabet (lower and uppercase mixed). It is 49 characters long at max. (2p)
Print back the string to the screen using only lowercase letters. (1p)
Print back the string to the screen using only uppercase letters. (1p)
Use your own procedure to print out only the last letter of the string to the screen. (3p)

Example run:

Give a string please... *aNAnAS*
ananas
ANANAS
S