**Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.**

Name: ________________________________

Date: ________________________________

| Question: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Points: | 15 | 10 | 10 | 5 | 5 | 5 | 10 | 10 | 10 | 15 | 95 |
| Score: | | | | | | | | | | | |

1. Describe exactly but briefly what is the output of the program.

```
#include <stdio.h>

double fun(double x, double y[]) {
    y[4] *= 2;
    printf("fun:    %f  %f\n", x, y[4]);
    return(y[0] + y[1] + y[2]);
}

int main(void) {
    double x, y[5];
    int i;
    x = 2.;
    for(i = 0; i < 5; i++) {
        y[i] = x*i;
    }
    printf("main 1: %f  %f\n", x, y[4]);
    x = fun(x, y);
    printf("main 2: %f  %f\n", x, y[4]);
    return(0);
}
```

(a) (5 points) The first printf statement in the main program:

________________________________________________

(b) (5 points) The printf statement in the function fun:

________________________________________________

(c) (5 points) The second printf statement in the main program:

________________________________________________

2. What is going to be printed by the fragment of the code below?

   (a) (5 points)

   (b) (5 points) Modify the value of variable ’i’ such that the last two digits printed are ’15’. (Use a red pen to indicate the changes in the code.)

```
int i = 41;
printf("%d    %o    %x\n", i, i, i);
```

3. (a) (5 points) Evaluate 01100001 $\oplus$ 00111001

   (b) (5 points) What are the decimal, the octal, and the hexadecimal values of the result?

**Multiple choice questions.**

For each of the following questions or statements, circle the number (0, 1, 2, or 3) of the one response that best answers the question or completes the statement.

4. (5 points) How many distinct values can be represented by a sequence of 3 bits?

   **0** 3

   **1** 4

   **2** 6

   **3** 8

5. (5 points) Which of the following is *not* a keyword in C?

   **0** for

   **1** if

   **2** loop

   **3** double

6. (5 points) Which of the following code fragments below produce identical output.

   1.
```
int x = 0;
printf("%d", x);
```

   2.
```
int x = 0;
printf("%d", 0);
```

   3.
```
int x = 0;
printf("%d", x++);
```

   4.
```
int x = 0;
printf("%d", ++x);
```

**0** only 1 and 2

**1** only 3 and 4

**2** only 1, 2, and 3

**3** all code fragments produce identical output

7. (10 points) Consider the code below.

```
int i, j;
for (i = 0; i < 5; i++)
    for (j = i; j < 5; j++)
        printf(”*”);
```

How many asterisks does this code print in total?

**0** 5

**1** 10

**2** 15

**3** 25

## One-time pad

8. (a) (5 points) Convert the numbers of your answers for Questions 4–7 to two-bit binary strings and concatenate four strings together.

The resulting binary string represents your one-byte plaintext message.

(b) (5 points) Encrypt the message using one-time pad given by the instructor.

## Stability of algorithms

9. As a part of the solution of a particular problem you need (repeated) calculations of the following expression:

$$\frac{1}{1-\sqrt{1-x}}. \tag{1}$$

for small $x$ such that $x \approx \epsilon$, where $\epsilon$ is machine epsilon.

(a) (5 points) Briefly describe what troubles you expect when using Eq. (2)?

(b) (5 points) Rewrite Eq. (1) to avoid those troubles.

10. As a part of the solution of a particular problem you need calculations (with the full machine precision) of the following integral

$$E_n = \int_0^1 x^{n+1} e^{x^2-1} dx, \tag{2}$$

where $n$ is an integer parameter. In your calculations $n$ takes few random values between 900 and 1000.

Integrating by parts you obtained the following recurrence relation.

$$E_n = \frac{1}{2}\left(1 - nE_{n-2}\right). \tag{3}$$

You also noticed that

$$E_0 = \frac{1}{2}\left(1 - \frac{1}{e}\right) \tag{4}$$

(a) (5 points) Write a fragment of memory-efficient C code that calculates $E_k$ for an even $k$ “naively” using Eqs. (3) and (4).

(b) (5 points) What is wrong with the “naive” approach?

(c) (5 points) Rewrite Eq. (3) to suggest a better algorithm.