

Programming for computerteknologi

Hand-in Assignment Exercises

Week 2: Development and Derivation of (simple) programs

Please make sure to submit your solutions **by next Monday**.

In the beginning of each question, it is described what kind of answer that you are expected to submit. If *Text and code answer* is stated, then you need to submit BOTH some argumentation/description and some code; if just (*Text answer*) or (*Code answer*) then just some argumentation/description OR code. The final answer to the answers requiring text should be **one pdf document** with one answer for each text question (or text and code question). When you hand-in, add a link to your GitHub repository in the beginning of your pdf file. Make sure that you have committed your code solutions to that repository.

Note: the **Challenge** exercises are *optional*, the others mandatory (i.e. you **have** to hand them in).

Exercises

- (1) (Old exam question. Solve it **without** running it, you will not be able to do so at the exam, Text answer)

Consider this program fragment

```
1  int a = ...; //see table
2  int b = ...; //see table
3  int c = (a - 1) * (a - 1);
4  int d = 0;
5  if(b == 0)
6      c = 1;
7  if(a == 0 || b == 0)
8      d = 1;
9  d = (c - 1) * (d - 1);
```

For the values of variables a and b as indicated below what is the resulting value of variable d?

a	b	d			
1	1	-1 <input type="checkbox"/>	0 <input type="checkbox"/>	1 <input type="checkbox"/>	2 <input type="checkbox"/>
1	0	-1 <input type="checkbox"/>	0 <input type="checkbox"/>	1 <input type="checkbox"/>	2 <input type="checkbox"/>
0	1	-1 <input type="checkbox"/>	0 <input type="checkbox"/>	1 <input type="checkbox"/>	2 <input type="checkbox"/>
0	0	-1 <input type="checkbox"/>	0 <input type="checkbox"/>	1 <input type="checkbox"/>	2 <input type="checkbox"/>

Tick the box next to the correct value for d with respect to the initial values of a and b as stated in the first two columns of the table.

- (2) (Text and code answer) Consider the following program:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int result;
6     int j;
7     int N;
8     scanf("%d", &N);
9     result = N;
10    j = 0;
11    while (j<N) {
12        result = result+1;
13        j = j+1;
14    }
15    printf("The result is %d.\n", result);
16    return 0;
17 }
```

What is the program calculating (i.e. what is the relation between the input and what is written in line 15)? You should not write in detail an explanation of what the different lines do, but what you would write as a comment in the top of the program describing the overall function of the program. Write a program that makes the same calculation but without a loop.

- (3) (Code answer) Write a program to compute the Fibonacci numbers. Your program should take an integer n as input from the user, and it then calculates and prints each Fibonacci number from the first Fibonacci number to the n^{th} . What is the pre condition? Remember to check it in your program. For example, a typical execution where the user inputs $n = 7$ could be:

```
7
0 1 1 2 3 5 8
```

- (4) (Code answer) Implement a *Linear Congruent Random Number Generator* computing:

$$x_{i+1} = (a * x_i + c) \% m$$

... where $m > 0$, $0 < a < m$, $0 < c < m$. The first value, called the *seed*, is x_0 . Your program should take four integer inputs from the user for a, m, c, x_0 . Given these input values from the user, your program should compute all values x_0, x_1, \dots, x_{m-1} and print them to the console as a table in the form:

```
n0  x0
n1  x1
...
nm-1 xm-1
```

Answer the following question: What pattern do you observe?

- (5) (Code answer) Write a program that computes the integer square root r of a number n such that $r^2 \leq n < (r+1)^2$. Your program should take integer n as input from the user and print the integer square root r . You are **not** allowed to use `math.h`. Remember to state the pre-condition and ensure that the program checks

for it being fulfilled. An example execution where the user inputs $n = 17$ could be:

17

4

- (6) (Text answer) What does the following program fragment compute (i.e. what is the relation between n and the value of y after the loop)? *Hint:* "run" the program by hand and observe the values of the variables as the program executes. You should **not** just run the program, but understand the different values that the variables holds as the program executes.

```
1 int y = 0;
2 int i = 0;
3 int j = 1;
4 int k = 6;
5 while (i < n)
6 {
7     y = y+j;
8     i = i+1;
9     j = j+k;
10    k = k+6;
11 }
```

- (7) (Text answer) When I asked ChatPGT the following question *What does the following program fragment compute?*

```
1 int x = 0;
2 int y = 0;
3 int h = b - a / 2;
4 while (x <= a)
5 {
6     printf("(%d,%d)", x, y);
7     if (h < 0)
8     {
9         h += b;
10    }
11    else
12    {
13        h += b - a;
14        y++;
15    }
16    x++;
17 }
```

I got the following answer:

The provided program fragment computes and prints the coordinates of points along a straight line segment using the Bresenham's Line Drawing Algorithm. The line segment starts from $(0, 0)$ and extends to (a, b) .

Here's how the code works step by step:

- Initialize x and y to 0, which represent the current point (x, y) on the line.
- Calculate h as $b - a / 2$. This value is used in the Bresenham's algorithm to determine the placement of the next point on the line.

- (c) Enter a `while` loop that continues as long as `x` is less than or equal to `a`. This loop iterates through the points on the line from $(0,0)$ to (a,b) .
- (d) Inside the loop:
 - i. Print the current point (x,y) using `printf`.
 - ii. Check if `h` is less than 0. If `h` is negative, it means that the next point on the line should move horizontally without changing the vertical position (`y`).
 - iii. If `h` is less than 0, execute the code block within the `if` statement.
 - iv. Increment `h` by `b`. This step accounts for moving horizontally by one unit.
 - v. If `h` is greater than or equal to 0, execute the code block within the `else` statement. This means that the next point on the line should move diagonally (incrementing both `x` and `y`).
 - vi. Increment `h` by `b - a`. This step accounts for moving diagonally by one unit.
 - vii. Increment `y` to change the vertical position of the current point.
 - viii. Increment `x` to move horizontally.

The result is that this code calculates and prints the coordinates of the points along the line segment from $(0,0)$ to (a,b) using the Bresenham's Line Drawing Algorithm. The algorithm ensures that the line is drawn accurately and efficiently with as few calculations as possible.

OpenAI. (2023). ChatGPT (August 3 Version) [Large language model]. <https://chat.openai.com>

Is ChatGPT's explanation correct? You need to

- find an explanation of *Bresenham's Line Drawing Algorithm* (REMEMBER to make a reference to where you have your information from, and argue that the source is reliable)
- explain the program using illustrations instead of the ChatGPT *read C program aloud* description
- argue (based on the explanation of the algorithm) that the program is correct
- make comments in the program to make it more readable

(8) Challenge

An alternative to calculating the square root is using Newton's method. Google Newton method for finding a square root and create a program that implements it. (you do not calculate the integer root but a real root)

- (9) **Challenge:** (PC-7.6.7) I collect marbles (colourful small glass balls) and want to buy boxes to store them. The boxes come in two types,

Type 1: each such box costs c_1 øre and can hold exactly n_1 marbles

Type 2: each such box costs c_2 øre and can hold exactly n_2 marbles

I want each box to be filled to its capacity, and also to minimise the total cost of buying them. Help me find the best way to distribute my marbles among the boxes.

Input The input begins with a line containing the integer n ($1 \leq n \leq 2,147,395,600$). The second line contains c_1 and n_1 , and the third line contains c_2 and n_2 . Here, c_1 , c_2 , n_1 , and n_2 are all positive integers having values smaller than 46,341.

Output Print a line containing the minimum cost solution (two nonnegative integers m_1 and m_2 , where m_i = number of type i boxes required if one exists. Otherwise print “failed”.

If a solution exists, you may assume that it is unique.

Example

```
43
1 3
2 4
13 1
```

```
40
5 9
5 12
failed
```

Hint Consider the unit costs for the two cases.