Report 7

Recursion

In this laboratory work I created a programm, which can do two tasks - recursion and drawing.

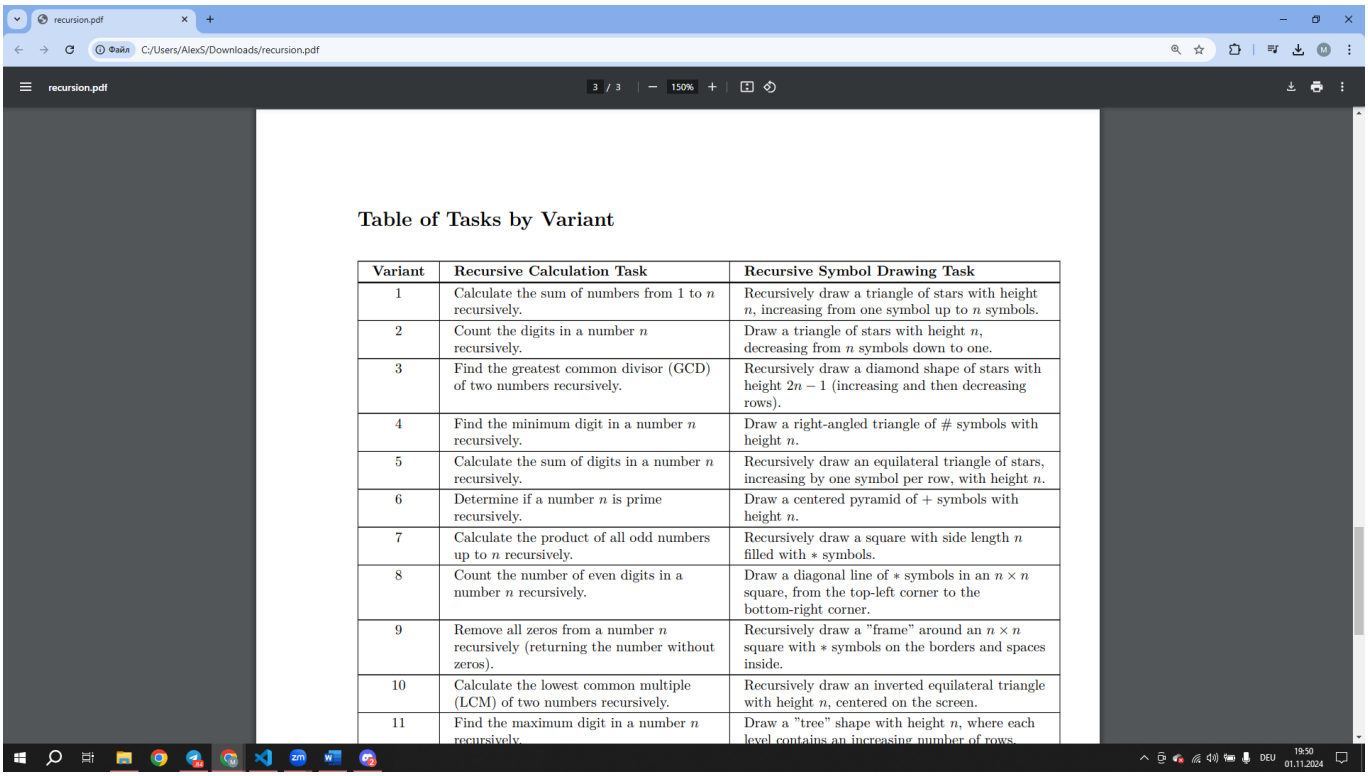

Table of Tasks by Variant

Variant	Recursive Calculation Task	Recursive Symbol Drawing Task
1	Calculate the sum of numbers from 1 to n recursively.	Recursively draw a triangle of stars with height n, increasing from one symbol up to n symbols.
2	Count the digits in a number n recursively.	Draw a triangle of stars with height n, decreasing from n symbols down to one.
3	Find the greatest common divisor (GCD) of two numbers recursively.	Recursively draw a diamond shape of stars with height $2n - 1$ (increasing and then decreasing rows).
4	Find the minimum digit in a number n recursively.	Draw a right-angled triangle of # symbols with height n.
5	Calculate the sum of digits in a number n recursively.	Recursively draw an equilateral triangle of stars, increasing by one symbol per row, with height n.
6	Determine if a number n is prime recursively.	Draw a centered pyramid of + symbols with height n.
7	Calculate the product of all odd numbers up to n recursively.	Recursively draw a square with side length n filled with $*$ symbols.
8	Count the number of even digits in a number n recursively.	Draw a diagonal line of $*$ symbols in an $n \times n$ square, from the top-left corner to the bottom-right corner.
9	Remove all zeros from a number n recursively (returning the number without zeros).	Recursively draw a "frame" around an $n \times n$ square with $*$ symbols on the borders and spaces inside.
10	Calculate the lowest common multiple (LCM) of two numbers recursively.	Recursively draw an inverted equilateral triangle with height n, centered on the screen.
11	Find the maximum digit in a number n recursively.	Draw a "tree" shape with height n, where each level contains an increasing number of rows.

Here is my task - option number 6.

Firstly I created a file with name "main.c", which had recursion calculation algorithm founded on "return" command.

Then I created file "mainOpt.c". I changed it main recursion algorithm to loop with command "for".

Here is th output of "main.c" file.

```
┌──(malex-kali㉿MA)-[~/Programming/Lab07]
└─$ gcc -g -O0 main.c -o expr

┌──(malex-kali㉿MA)-[~/Programming/Lab07]
└─$ gdb ./expr
GNU gdb (Debian 13.2-1+b2) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later http://gnu.org/licenses/gpl.html
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

```
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
--Type for more, q to quit, c to continue without paging--c
https://www.gnu.org/software/gdb/bugs/.
Find the GDB manual and other documentation resources online at:
http://www.gnu.org/software/gdb/documentation/.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./expr...
(gdb) r
Starting program: /home/malex-kali/Programming/Lab07/expr
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/ libthread_db.so.1".
TRUE
```

```
          +
         +++
        +++++
       +++++++
      +++++++++
     ++++++++++
    ++++++++++++
   ++++++++++++++
  ++++++++++++++++
 ++++++++++++++++++
++++++++++++++++++++
```

```
++++++++++++++++++++++
[Inferior 1 (process 7892) exited normally]
(gdb) q
```

Output for "mainOpt.c" file

```
┌──(malex-kali㉿MA)-[~/Programming/Lab07]
└─$ gcc -g -O0 mainOpt.c -o exprOpt

┌──(malex-kali㉿MA)-[~/Programming/Lab07]
└─$ gdb ./exprOpt
GNU gdb (Debian 13.2-1+b2) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later http://gnu.org/licenses/gpl.html
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
```

> Type "show configuration" for configuration details.
> For bug reporting instructions, please see:
> --Type for more, q to quit, c to continue without paging--c
> https://www.gnu.org/software/gdb/bugs/.
> Find the GDB manual and other documentation resources online at:
> http://www.gnu.org/software/gdb/documentation/.
>
> For help, type "help".
> Type "apropos word" to search for commands related to "word"...
> Reading symbols from ./exprOpt...
> (gdb) r
> Starting program: /home/malex-kali/Programming/Lab07/exprOpt
> [Thread debugging using libthread_db enabled]
> Using host libthread_db library "/lib/x86_64-linux-gnu/ libthread_db.so.1".
> TRUE
>
> ---
>
> ```
>            +
>           +++
>          +++++
>         +++++++
>        +++++++++
>       ++++++++++
>      ++++++++++++
>     +++++++++++++
>    +++++++++++++++
>   +++++++++++++++++
>  +++++++++++++++++++
> ```
>
> ++++++++++++++++++++++
> [Inferior 1 (process 7892) exited normally]
> (gdb) q

These two outputs were generated with input value "11". Then, to check what is th difference in speed between these two versions of programms, I put "74789380547" as input value, disconnected drawing function and tested time.

For "main.c" it is the next:

> real 0m0.002s
> user 0m0.002s
> sys 0m0.000s

And for "mainOpt.c" it is:

> real 0m0.001s
> user 0m0.001s
> sys 0m0.000s