



# B2- Synthesis Pool

---

B-ADM-144

## Palindrome

---

Iterated Turning Over Palindromes

v1.0



# Requirement

## `my_factrec_synthesis`, `my_squareroot_synthesis`

Write a c file named `requirement.c` that contains the following two functions:

### `my_factrec_synthesis`

Write a recursive function that will return the factorial of the number given as a parameter. It should be prototyped the following way:

```
int my_factrec_synthesis(int nb)
```

In case of error, the function should return 0.

### `my_squareroot_synthesis`

Write a function that returns the square root (if it is a whole number) of the number given as argument. If the square root is not a whole number, the function should return -1. It should be prototyped the following way:

```
int my_squareroot_synthesis(int nb);
```



These functions must produce an answer in under 2 seconds.  
Overflows must be handled (as errors).



Only `malloc` and `free` are allowed from `libc`.  
The rest of the project will not be corrected unless this requirement is fully functional (and rewritten).



The file must be placed at the root of your git repository.  
It will be compiled with our main function, and our Makefile (the `-I` flag being empty).



# Palindrome

## Iterated Turning Over Palindromes

---

binary name: palindrome  
repository name: ADM\_palindrome\_\$YEAR  
repository rights: ramassage-tek  
language: C  
group size: 1  
compilation: via Makefile, including re, clean and fclean rules

---



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

### Authorized functions: every functions from the libC

A palindrome is a word or a number which reads the same backward or forward (that is for numbers, numbers invariant when reversed).

A  $n$ -iteration turnover palindrome is a number which leads to a palindrome after  $n$  iterations of the following procedure:

1. reverse the number,
2. add both the number and its reversal to obtain a new number,
3. iterate from the first step until you get a palindromic number.

For instance: 121 is a 0-iteration turnover palindrome, 123 is a 1-iteration reversal palindrome and 4782 is a 3-iteration turnover palindrome.

The goal of this project is to find such iterated reversal palindromes, and the smallest number of iterations, in order to get a palindromic number, given the constraints on this number of iterations.

```
Terminal
~/B-ADM-144> ./palindrome -h
USAGE
    ./palindrome -n number -p palindrome [-b base] [-imin i] [-imax i]
DESCRIPTION
    -n nb      integer to be transformed ( $\geq 0$ )
    -p pal     palindromic number to be obtained (incompatible with the  $-n$  option) ( $\geq 0$ )
               if defined, all fitting values of  $n$  will be output
    -b base    base in which the procedure will be executed ( $1 < b \leq 10$ ) [def: 10]
    -imin i    minimum number of iterations, included ( $\geq 0$ ) [def: 0]
    -imax i    maximum number of iterations, included ( $\geq 0$ ) [def: 100]
```



Your program output has to be strictly identical to the one below.



All the numbers are displayed in base 10.

If there is no solution, the message "no solution\n" is printed on the standard output, but is not considered as an error.

```
Terminal
~/B-ADM-144> ./palindrome -n 4782 toto
invalid argument
```

```
Terminal
~/B-ADM-144> ./palindrome -n 4782 -b 10
4782 leads to 46464 in 3 iteration(s) in base 10
```

```
Terminal
~/B-ADM-144> ./palindrome -n 64 -b 2
64 leads to 65 in 1 iteration(s) in base 2
```

```
Terminal
~/B-ADM-144> ./palindrome -p 363 -b 10
3 leads to 363 in 6 iteration(s) in base 10
6 leads to 363 in 5 iteration(s) in base 10
12 leads to 363 in 4 iteration(s) in base 10
15 leads to 363 in 3 iteration(s) in base 10
21 leads to 363 in 4 iteration(s) in base 10
24 leads to 363 in 3 iteration(s) in base 10
30 leads to 363 in 4 iteration(s) in base 10
33 leads to 363 in 3 iteration(s) in base 10
39 leads to 363 in 2 iteration(s) in base 10
42 leads to 363 in 3 iteration(s) in base 10
48 leads to 363 in 2 iteration(s) in base 10
51 leads to 363 in 3 iteration(s) in base 10
57 leads to 363 in 2 iteration(s) in base 10
60 leads to 363 in 3 iteration(s) in base 10
66 leads to 363 in 2 iteration(s) in base 10
75 leads to 363 in 2 iteration(s) in base 10
84 leads to 363 in 2 iteration(s) in base 10
93 leads to 363 in 2 iteration(s) in base 10
132 leads to 363 in 1 iteration(s) in base 10
231 leads to 363 in 1 iteration(s) in base 10
330 leads to 363 in 1 iteration(s) in base 10
363 leads to 363 in 0 iteration(s) in base 10
```



```
Terminal
~/B-ADM-144> ./palindrome -p 363 -b 10 -imax 2
39 leads to 363 in 2 iteration(s) in base 10
48 leads to 363 in 2 iteration(s) in base 10
57 leads to 363 in 2 iteration(s) in base 10
66 leads to 363 in 2 iteration(s) in base 10
75 leads to 363 in 2 iteration(s) in base 10
84 leads to 363 in 2 iteration(s) in base 10
93 leads to 363 in 2 iteration(s) in base 10
132 leads to 363 in 1 iteration(s) in base 10
231 leads to 363 in 1 iteration(s) in base 10
330 leads to 363 in 1 iteration(s) in base 10
363 leads to 363 in 0 iteration(s) in base 10
```

```
Terminal
~/B-ADM-144> ./palindrome -p 363 -b 10 -imin 6
3 leads to 363 in 6 iteration(s) in base 10
```

```
Terminal
~/B-ADM-144> ./palindrome -p 363 -b 10 -imin 7
no solution
```