StrongPy

Software Documentation

Author: matiwa

Table of contents

Table of contents…………………………………………………………………………….2

Introduction……………………….…………………………………………………...3

Describing of the application's operation……………………………………………...3

What is needed for use?..3

Algorithm used………………..………………………………………………………….3

Source code description………………………………………………………………..5

List of drawing...…………………………………………………………………………….5

List of listings……………………………………………………………………………5

Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. The application is used to calculate the strong of a selected non-negative integer n.

Describing of the application’s operation

When you run the script, the application prompts you to enter n. There is no protection for a negative value. In case of a mistake made by the user, the program will display the result as for 0 and 1 equal to 1.

What is needed for use?

The application does not require installation. It only needs the Windows operating system.

Algorithms used

The only algorithm used in application development is to calculate the strong of any non-negative and integer number. For n equal to 0, the strong is 1. If n is greater than or equal to 1, the strong for 0 will be 1 multiplied by the series of numbers preceding n and then itself. An example for the numbers n of 4, 5, and 6 is below.

n = 4

n! = 1 * 2 * 3 * 4   4! = 24

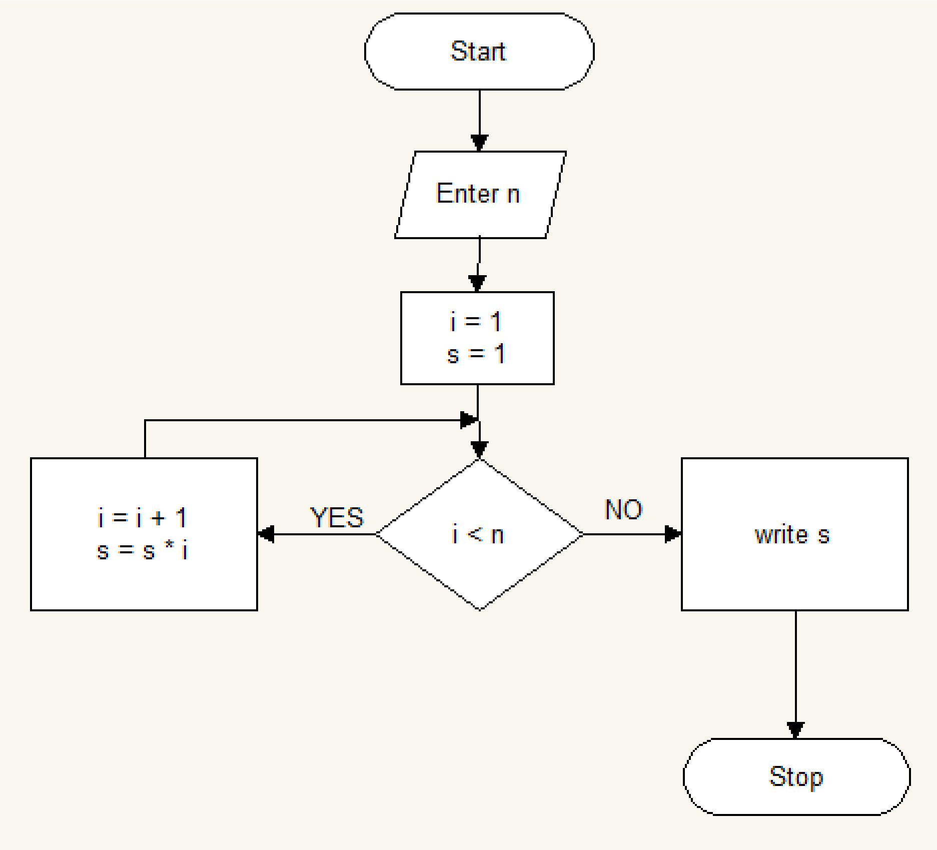
n = 5

n! = 1 * 2 * 3 * 4 * 5   5! = 120

n = 6

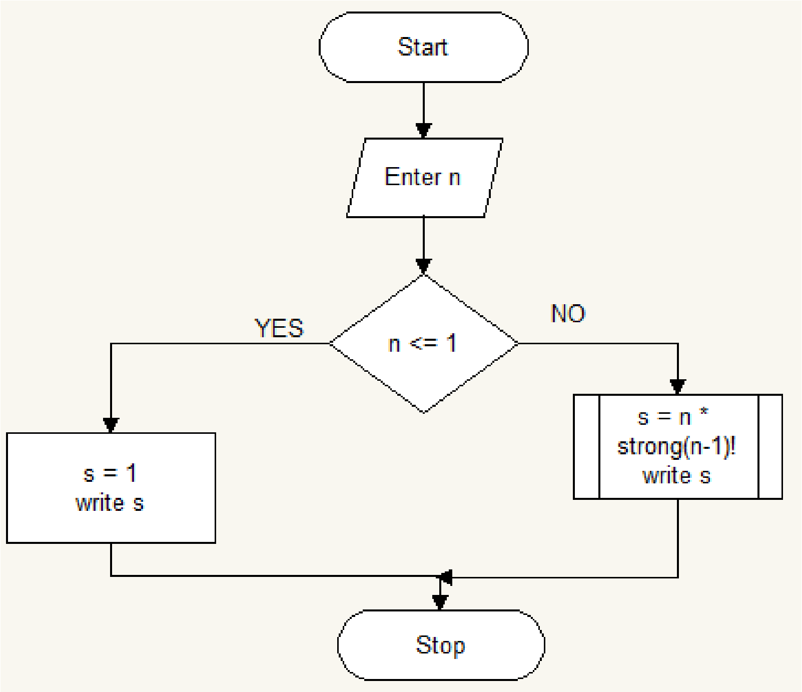
n! = 1 * 2 * 3 * 4 * 5 * 6   6! = 720

There are two ways to compute strong. It can be iterative and recursive. This is the iterative version in the form of a block diagram.



Drawing 1: Interative version [own study]

The recursive version is slightly different.



Drawing 2: Recursive version [own study]

Source code description

The project was made in the Python programming language, in the PyScripter programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
n=int(input("Enter n: "))
i=1
j=1

while (i<n):
    i+=1
    j*=i
print(str(n)+"!="+str(j))
```

Listing 1: Source code [own study]

List of drawings

Drawing 1: Interative version [own study]……………………………………………………4

Drawing 2: Recursive version [own study]……………………………………………………4

List of listings

Listing 1: Source code [own study]…………………………………………………………5