



Sofya Programming Language

Version 1.0 User Manual

Table of Contents

Introduction.....	4
The Life of Sofya Kovalevskaya.....	4
The Basics of Programming.....	5
Introduction.....	5
Program Flowcharts.....	6
Setting up Sofya Version 1.0.....	7
Using the Sofya Version 1.0 IDLE.....	8
Introduction.....	8
Parts of the Sofya Version 1.0 IDLE.....	9
Filename Bar.....	9
Program Editor.....	9
Input Box.....	10
Program's Output.....	10
Menu Bar.....	10
The Run Menu.....	11
The File Menu.....	11
The Sofya Rules of Thumb.....	12
Programming Using Sofya.....	12
Writing Data Types.....	12
Writing One Data Type per line.....	12
Example.....	13
Solution.....	13
Writing Many Data Types on the same line.....	15
Using Operations.....	15
Operator Precedence.....	16
Example.....	17
Solution.....	17
Finding the r^{th} Root of a number.....	19
Example.....	20
Solution.....	20
Commenting.....	21
Example.....	21
Solution.....	21
Using Variables.....	23
Example.....	23
Solution.....	23
Rules about Variables.....	25
Receiving Input.....	26
Example.....	26
Solution.....	26
Receiving Special Input.....	30
E-Form.....	31
If...Then Statements.....	31

Comparison Operators.....	31
Example.....	32
Solution.....	33
Rules about If...Then Statements.....	36
Constants.....	37
Example.....	38
Solution.....	38
If...Then...Else Statements.....	40
Example.....	41
Solution.....	41
Rules about If...Then...Else Statements.....	44
Using Ranges.....	45
Rules of Ranges.....	45
If...Otherwise-If...Else Statements.....	46
Example.....	46
Solution.....	47
Rules about If...Otherwise-If...Else Statements.....	50
Nested If Statements.....	51
Example.....	51
Solution.....	51
Rules of Nested If Statements.....	54
Static Variable Operations (SVOs).....	54
Do This...Until Loops.....	56
Example.....	56
Solution.....	57
Rules about Do This...Until Loops.....	60
Breaking Loops.....	61
Example.....	61
Solution.....	61
Infinite Loops.....	64
Example.....	64
Solution.....	65
Nested Do This...Until Loops.....	67
Example.....	67
Solution.....	67
Program Errors in Sofya.....	72
Table of Program Errors.....	72
Loop Error.....	73
IDLE Crash.....	73
References used in developing Sofya.....	74
Conclusion.....	75

Introduction

Welcome to the world of programming! In this manual, we assume that this is your first time programming. Thank you for choosing to start programming using Sofya! Sofya is a **programming language** that was named after a female mathematician called **Sofya Kovalevskaya**. Sofya was made by **Timothy Oywera** using another programming language called **Python**. Sofya was made to enable people to program computers to solve **mathematical** and **scientific problems**. Before we learn about how to use the Sofya Programming Language, we are going to look at **the life of Sofya Kovalevskaya** and the **basics of programming**.

The Life of Sofya Kovalevskaya

- ❖ She was born in **Russia** in a town called **Moscow** in **1850**.
- ❖ She became interested in **Mathematics** and **Physics** from an **early age**.
- ❖ She taught herself **trigonometry** when she was **fourteen years old**.
- ❖ However, her father was not happy with her studying Physics and Mathematics, so he took her textbooks away.
- ❖ After she finished high school, she wanted to **continue** her **studies** in **University**.
- ❖ However, at that time **Russian Universities** would **not** admit **women** but there was a University in **Switzerland** that admitted women. So Sofya married **Vladimir Kovalevsky** so that she could travel to Switzerland.
- ❖ Later she moved to **Germany** to get a **PhD**.
- ❖ In **1874**, she presented reports about: **partial differential equations**, **Saturn's rings** and **elliptic integrals**.
- ❖ In the report about partial differential equations she had written about a law called **The Cauchy-Kovalevskaya Theorem**.
- ❖ Later she returned to **Russia** and she gave birth to her **daughter** in **1878**.
- ❖ She **divorced** her **husband** in **1881**.
- ❖ She became a **lecturer of Mathematics** in the year **1883** in the **University of Stockholm**.
- ❖ In **1890**, she wrote a book called **Memories of Childhood**.
- ❖ She **died** in **1891** at the age of **41 years old** because of **influenza**.

- ❖ One of her famous quotes was, "**Many who have not studied Mathematics confuse this science with arithmetic and consider it dry and soulless. However, it is a science that requires great imagination.**"



A picture of Sofya Kovalevskaya

The Basics of Programming

Introduction

- **Programming** is giving instructions to a computer if you want it to do something for you.
- The instructions that a computer is given are called **programs**.
- When people are programming computers, they use **programming languages**.
- The rules of a programming language are called **syntax**.
- People who program computers are called **Computer Programmers**.
- A **string** in programming is anything that is inside **speech marks (" ")** — for example, "**Tinyboxyboy**", "**Introduction to Sofya Programming**" or "**My £150,000**". When you make something a string in programming, it helps the

Have fun using Sofya Version 1.0!

computer know that the string is not a **reserved word**. A reserved word is a word that already has a meaning in a programming language, for example, in Sofya when you use the word **write** (without speech marks), the computer knows that you want it to write something on the screen but if you say "**write**" (with speech marks), the computer now knows that "write" is just a normal word and not the reserved word for writing things on the screen.

- An **expression** in programming is a calculation — for example, **23 x 56**.

Program Flowcharts

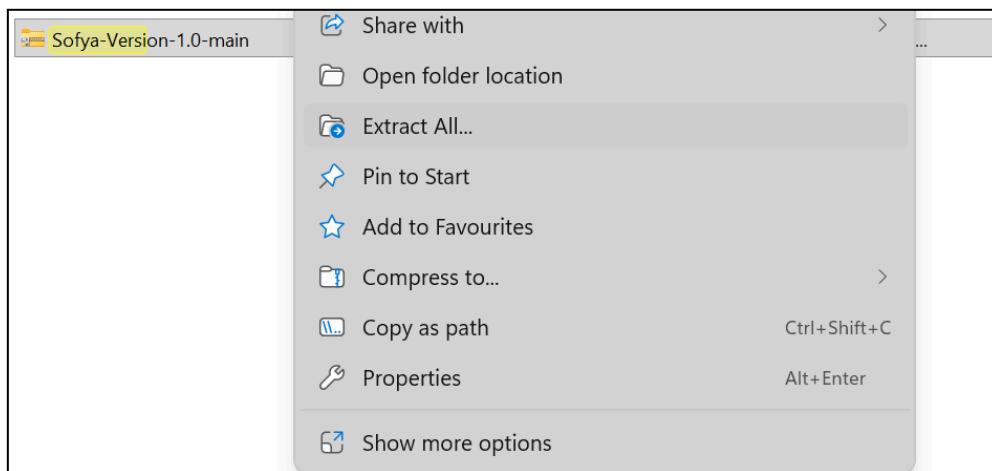
People usually draw **program flowcharts** to help them understand **programs**. In this manual, sometimes we will also use program flowcharts to help you understand programs. Here are the symbols that we will use in the program flowcharts and their meanings:

Symbol	Name	Function
	Terminator	It shows the start or end of a program
	Process	It shows how the computer uses data or how the computer does calculations
	Input or Output	It shows that the person using the computer gives the computer data(input) or the computer shows a person something on the screen(output)
	On-page Connector	It connects flowcharts that are on the same page
	Off-page Connector	It connects flowcharts that are on different pages
	Decision	It shows that the computer has to make a choice
	Flow Lines	It shows the next step of a program

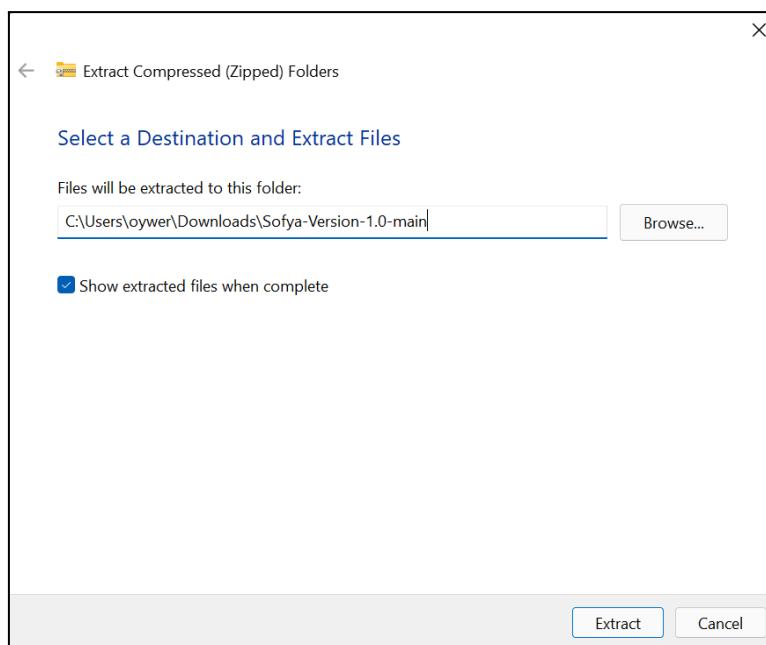
Setting up Sofya Version 1.0

To set up Sofya Version 1.0, please follow these steps:

1. Click this link to go to **GitHub** and then you can clone the Sofya Version 1.0 repository: <https://github.com/oyweraa-hue/Sofya-Version-1.0>
2. If you cloned the Sofya Version 1.0 repository (in step 1), you should find it in the **downloads section** on your computer (it will be in a folder named **Sofya-Version-1.0-main**). You can unzip (extract the files from) this folder by right clicking on it then choosing the option for extracting the files.



3. The computer will ask you where you want to extract the files to. Choose the place where you want the computer to extract the files to.



4. Then go to the place where the computer extracted the files to and you will find a folder called **Sofya-Version-1.0-main**.

Have fun using Sofya Version 1.0!

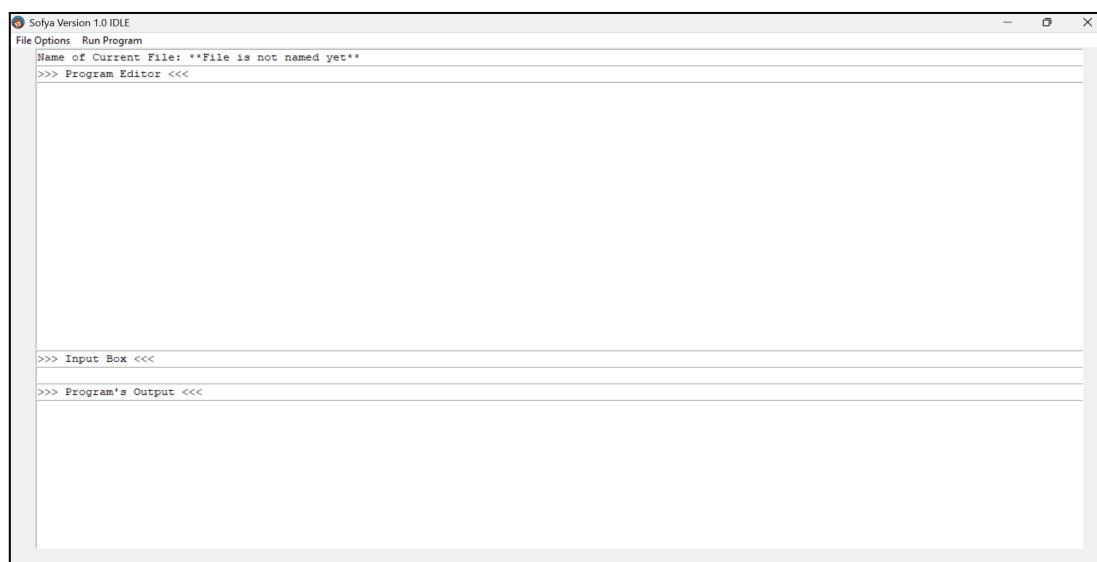
5. If you click on this folder, you will see three other folders called **Sofya Version 1.0 Documentation**, **Sofya Version 1.0 for Windows** and **Sofya Version 1.0 for MacOS**. The *Sofya Version 1.0 Documentation* folder has the *manual*, the *README* and the *licence* for Sofya. The *Sofya Version 1.0 for Windows* folder has the *Sofya Version 1.0 IDLE* (that can only work if you are using Windows Operating System). Lastly, the *Sofya Version 1.0 for MacOS* folder has the *Sofya Version 1.0 IDLE* (that can only work if you are using Macintosh Operating System).

Name	Status	Date modified	Type
📁 Sofya Version 1.0 Documentation	✓	13 Jan 2026 08:10	File folder
📁 Sofya Version 1.0 for MacOS	✓	13 Jan 2026 09:57	File folder
📁 Sofya Version 1.0 for Windows	✓	31 Dec 2025 12:53	File folder

Using the Sofya Version 1.0 IDLE

Introduction

The **Sofya Version 1.0 Integrated Development Learning Environment (IDLE)**, is a place where you can write your Sofya programs and you can also see how your Sofya programs work.



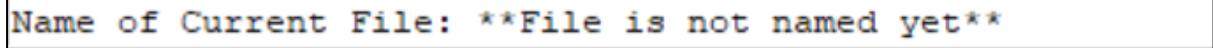
A screenshot of the Sofya Version 1.0 IDLE

Have fun using Sofya Version 1.0!

Parts of the Sofya Version 1.0 IDLE

Here is an explanation of all the parts of the Sofya Version 1.0 IDLE and their functions.

Filename Bar

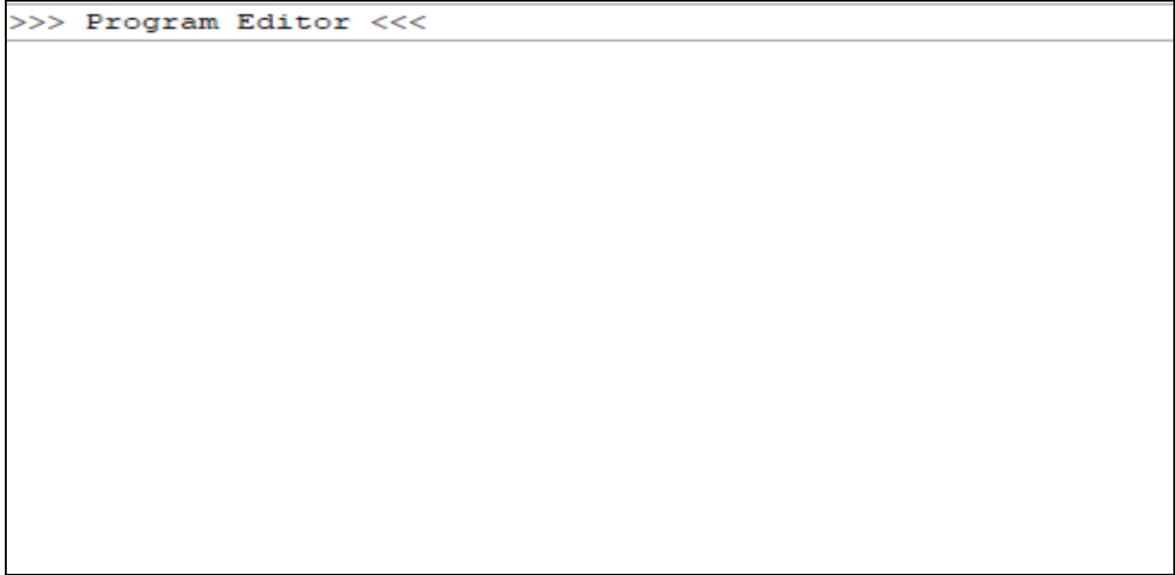


```
Name of Current File: **File is not named yet**
```

A screenshot of the Filename Bar

It tells you the **name** of the Sofya file that you are using (A **Sofya file** is also called a **Sofya program**). If you have not named your Sofya file, the computer will tell you that the file has not been named yet.

Program Editor



```
>>> Program Editor <<<
```

A screenshot of the Program Editor

It is the place where you write your Sofya program. You can always scroll up, down, left or right, in the program editor, to see the Sofya program that you have written if it is not able to fit on the screen.

Input Box



A screenshot of the Input Box

Sometimes when you are programming, you might want the computer to ask the person, who is using the computer, some **questions**. The person, who is using the computer, will **answer** these questions in the input box.

Program's Output



A screenshot of the Program's Output

It is the place where you see what your Sofya program is **doing**. You can always scroll up, down, left or right, in the program's output, to see what the Sofya program is doing if it is not able to fit on the screen.

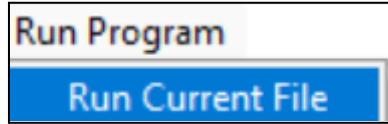
Menu Bar



A screenshot of the Menu Bar

The menu bar has two parts — the **file menu** and the **run menu**.

The Run Menu

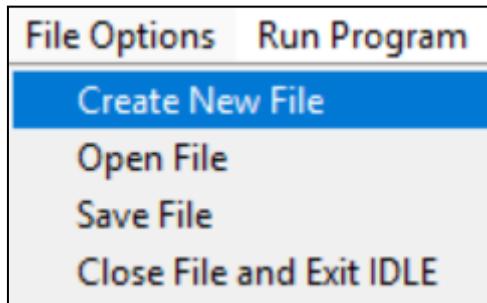


A screenshot of the Run Menu

It tells the computer to do the program (instructions) that you have given it in the program editor. When you use the run menu, we say that you are **running the file** (or **program**). Every time you run a program, the computer saves the file for you automatically.

Note: The computer will save the file for you automatically, only if you **save your file**. If you do not save your file, the computer will show you a **File Not Saved Error**.

The File Menu



A screenshot of the File Menu

The file menu is used to do different things to a Sofya program (Sofya file). It has **four** choices:

- **Create New File:** It is used if you want to make a new Sofya program. If you try to make a new Sofya program and you are already using a new Sofya program (which has not yet been given a name), the computer will show you a **New File Error**.
- **Open File:** It is used if you want to open a Sofya program (Sofya file) that you had already made before.
- **Save File:** It is used to save a Sofya program (Sofya file) that you have made. When you save a Sofya program, it is also a way of giving a Sofya file a name.
- **Close File and Exit IDLE:** It is used to exit (or close) the Sofya Version 1.0 IDLE. It is a good habit to save your Sofya file before you exit the IDLE.

The Sofya Rules of Thumb

Here are some basic rules which will help you when using Sofya:

1. Sofya is **not** a case sensitive programming language.
2. When you are typing **numbers** (that are not strings) in the Sofya Version 1.0 IDLE **program editor**, you should **not separate** the digits of the number using a comma. For example, if you want to type the number **one million**, in the program editor, you should type it as **1000000** instead of **1,000,000**.
3. When you are typing numbers in the Sofya Version 1.0 IDLE **input box**, you **can separate** the digits of a number using a comma (if you want). For example, if you want to type the number **two million**, in the input box, you can type it as **2,000,000** or **2000000**.
4. Before starting to write a Sofya program, it is a good habit to start with the words "**Start this program**" (But the program will still work even if you do not say "**Start this program**").
5. After you finish writing a Sofya program, it is a good habit to end with the words "**Stop this program**" (But the program will still work even if you do not say "**Stop this program**").
6. Sofya is designed to be used for **Mathematics** and **Science** and so that is why it has many features for Mathematics and Science.

Programming Using Sofya

Writing Data Types

Writing One Data Type per line

Sometimes, you might want the computer to **write** something on the screen. This is called **writing**. When you want to write something using Sofya you should use a function called **Write**. There is a table, on the next page, showing you how to write different data types in Sofya:

Data Type	Example	How to write it using Sofya
String	Hello World	Write "Hello World" on the screen
Number	6829	Write 6829 on the screen
Expression	1+3	Write 1+3 on the screen
Variable	Energy	Write Variable[Energy] on the screen
Constant	Pi	Write Constant[Pi] on the screen

Example

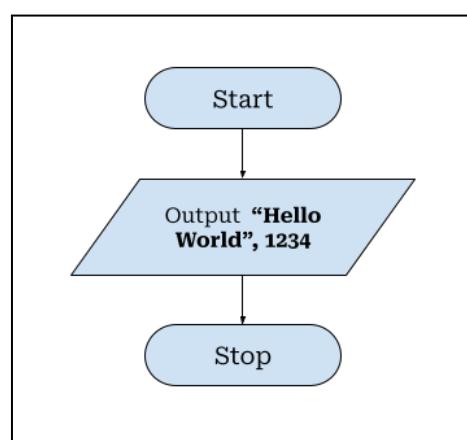
Use Sofya to write "**Hello World**" and the number **1234**.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

```
Start this program
    Write "Hello World" on the screen
    Write 1234 on the screen
Stop this program
```

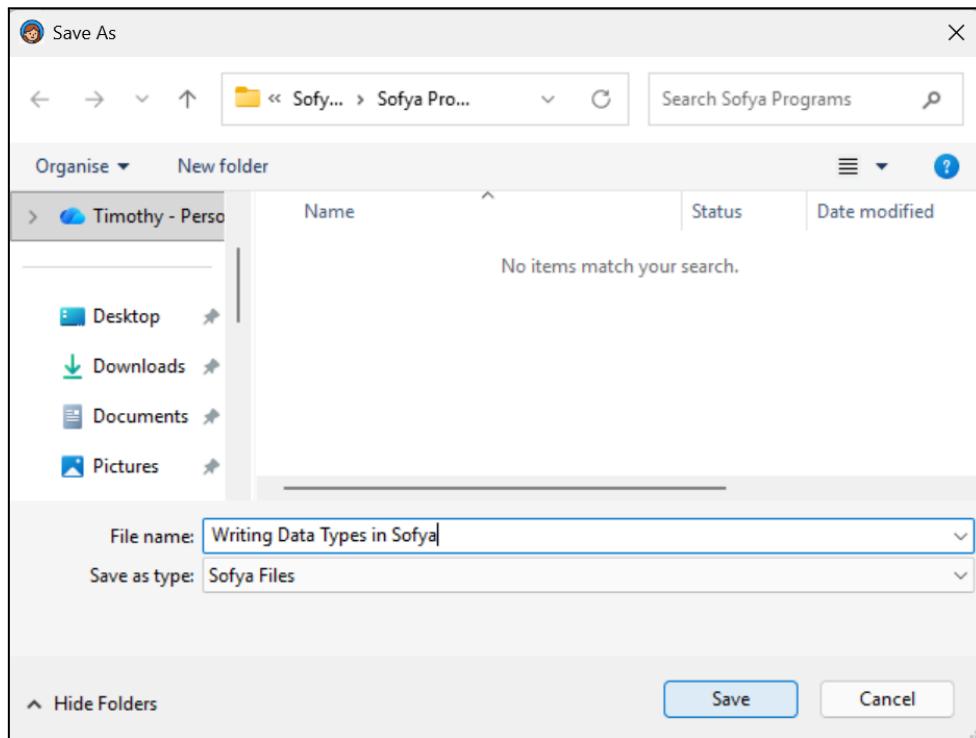
2. Here is a program flowchart for this program:



3. Here is an explanation for the program (on the next page).

- **Line 1:** We are telling the computer that we want to start programming
(The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to write "**Hello World**".
- **Line 3:** We are telling the computer to write "**1234**".
- **Line 4:** We are telling the computer that we want to stop programming
(The program will still work even if you do not say "Stop this program").

4. Save this Sofya program.



5. Run the Sofya program.
6. The computer will write "**Hello World**" and "**1,234**" in the program's output.

```
>>> Program's Output <<<  
  
>>> THE PROGRAM HAS STARTED...  
  
HELLO WORLD  
1,234.0  
  
>>> THE PROGRAM HAS ENDED
```

A screenshot of a terminal window showing the output of a Sofya program. The text is color-coded: blue for command-like text ('>>>'), black for the program's output ('THE PROGRAM HAS STARTED...', 'HELLO WORLD', '1,234.0'), and red for the final message ('THE PROGRAM HAS ENDED'). The background of the terminal is light gray.

Writing Many Data Types on the same line

In Sofya, you are also allowed to write many data types on the screen in the same line. However, if you decide to do this, you **cannot** write a **number** or **expression** on the screen — but you are still **allowed** to write **strings**, **variables** and **constants** on the screen. If you have to write a number or expression on the screen, you can write them as a **string**, for example, "1234".

Let us say that you want to write **two strings** and **one constant** on the screen in the same line. You can write a Sofya program like this:

- **Line 1:** Start this program
- **Line 2:** Write "The speed of light is" and also write Constant[c] and also write "m/s" on the screen
- **Line 3:** Stop this program

In Sofya, when you want to connect (or join) the different data types that you want the computer to write on the screen, you say, "**and also write**". So, we can say that the general way of writing many data types on the screen in the same line is:

Write { data type 1 } and also write { data type 2 } and also write { data type 3 } ... and also write { data type n } on the screen

Using Operations

Sometimes, you might want to do **calculations** using **operations**. Here are the operations that you can use in Sofya:

Name of the operation	How it looks in Sofya	Function	Example
Addition	+	It means that a number is added with another number	$1+1 = 2$
Subtraction	-	It means that a number is subtracted from another number	$2-2 = 0$

Multiplication	x	It means that a number is multiplied with another number	$3 \times 3 = 9$
Division	/	It means that a number is divided with another number	$4 / 4 = 1$
Brackets or Parentheses	()	It is used to do some operations first	$(8+3) \times 2 = 22$
Exponentiation	Exp	An exponent is a small number that is on top of another number. For example, if you write 2^5 , the small 5 that is on top of 2 is an exponent	$2 \text{ Exp } 3 = 8$
Floor	Floor	Flooring is when you tell a computer to divide a number with another number, but the computer will not show you the remainder	$7 \text{ Floor } 3 = 2$
Modulus	Modulus	Modulus is when you tell a computer to divide a number with another number, but the computer will only show you the remainder	$7 \text{ Modulus } 3 = 1$

Operator Precedence

Sometimes, when you want to use operations, they can be a lot. For example, let us say that you want Sofya to do this calculation :

$$10 + 25 \times (34 - 4) \text{ EXP } 6$$

In that calculation, there are many operators (there is addition, multiplication, subtraction, parentheses and exponentiation). Because of this, Sofya uses something called **operator precedence**, so that Sofya will be able to know how to do

calculations that have many operations. Operator precedence helps Sofya to know which operation it should do first.

This is a small chart that is showing us the operator precedence for Sofya :

Parenthesis or Brackets (*It is done first*)



Exponentiation



Division or Floor or Modulus



Multiplication



Addition



Subtraction (*It is done last*)

Example

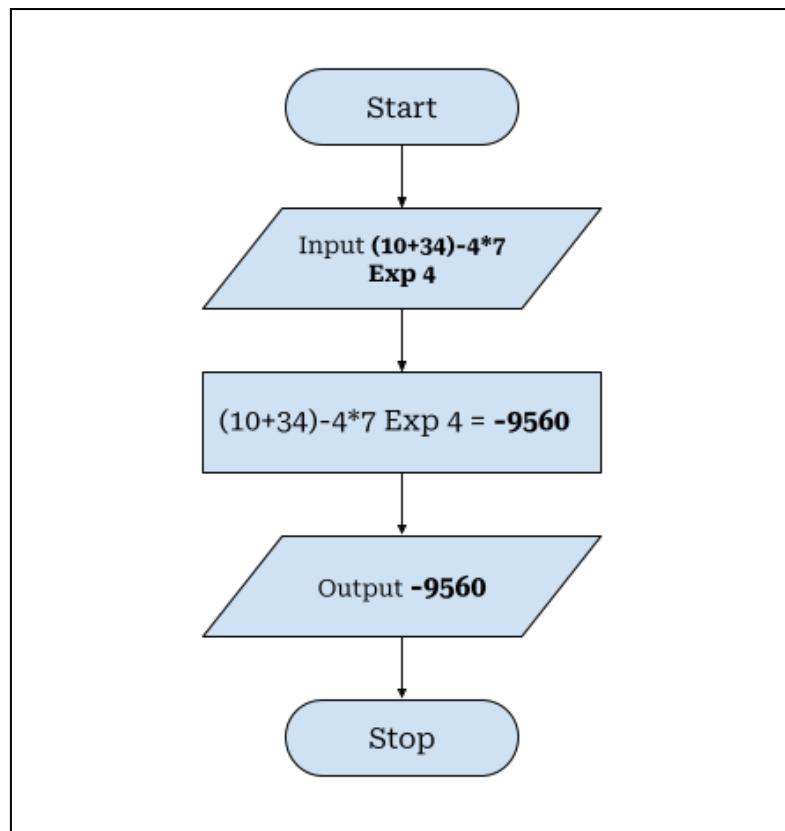
Use Sofya to calculate $(10 + 34) - 4 \times 7^4$.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

```
Start this program
Write (10+34)-4x7 Exp 4 on the screen
Stop this program
```

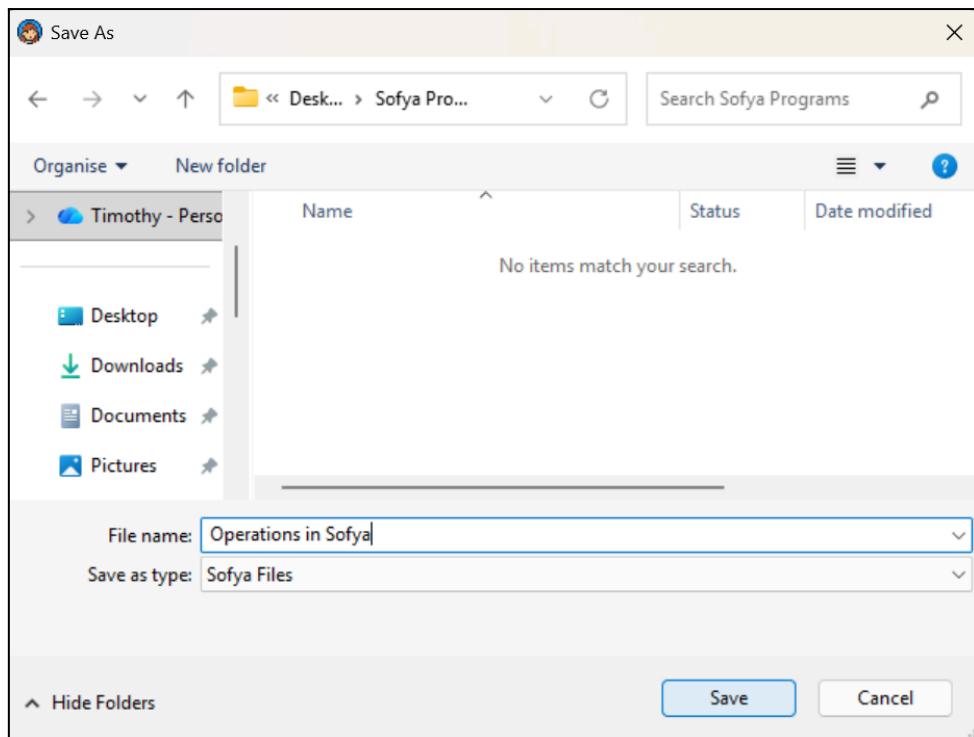
2. On the next page there is a program flowchart for this program:



3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to write "**(10+34)-4x7 Exp 4**". When you tell Sofya to write an expression, the computer will calculate the expression first and then the computer will write the answer of the expression.
- **Line 3:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



5. The computer will write "-9,560.0" on the program's output.

```
>>> Program's Output <<<
>>> THE PROGRAM HAS STARTED...
-9,560.0
>>> THE PROGRAM HAS ENDED
```

Finding the r^{th} Root of a number

Let us say that n and x are numbers. If we say that x is the r^{th} root of n , it means that we can multiply x by itself a certain number of times to get n . For example, if we say that 3 is the 4^{th} root of 81, it means that we can multiply 3 by itself 4 times to get 81 ($3 \times 3 \times 3 \times 3 = 81$ or $3^4 = 81$).

In mathematics there are many methods of finding the r^{th} root of a number, but the one that is used in Sofya is called the **index method** or **power method**. The formula for the index method is: $x = n^{\frac{1}{r}}$, where:

- n is the number that you want to find the r^{th} root of.
- x is the r^{th} root of n .
- r is the type of root that you want to calculate, for example, **square root**, **cube root**, **8th root**, etc.

For example, if we want to find the:

- **Square root of 4**, we can calculate it by doing $4^{\frac{1}{2}} = 2$
- **Cube root of 9**, we can calculate it by doing $9^{\frac{1}{3}} = 2.080$ (to 3 d.p)
- **8th root of 16**, we can calculate it by doing $16^{\frac{1}{8}} = 1.414$ (to 3 d.p)

Example

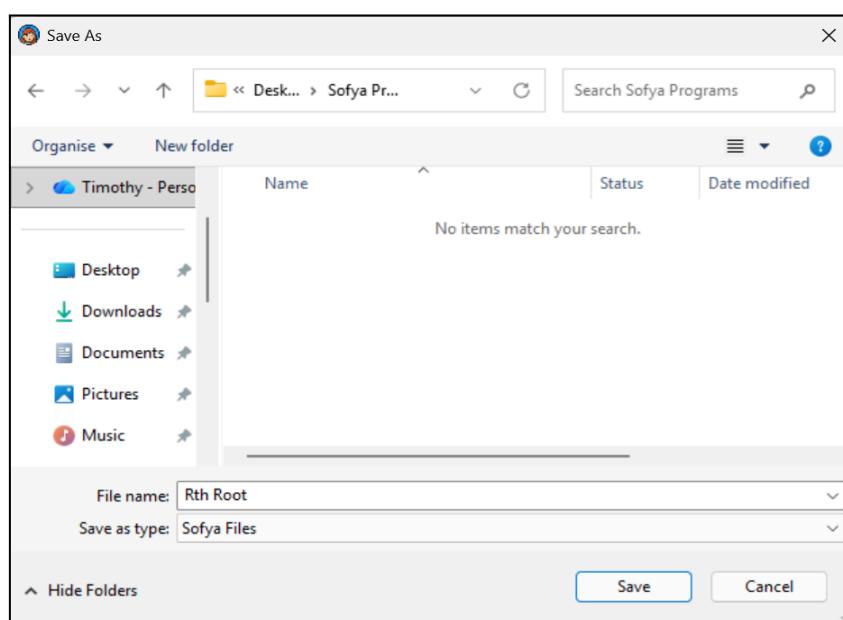
Use Sofya to calculate the 27th root of 89,456.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

```
Start this program
Write 89456 exp (1/27) on the screen
Stop this program
```

2. Save this Sofya program and then run it.



3. The computer will write "1.525432319222137" on the program's output.

```
>>> Program's Output <<<  
  
>>> THE PROGRAM HAS STARTED...  
  
1.525432319222137  
  
>>> THE PROGRAM HAS ENDED
```

Commenting

Sometimes, you might want to explain to someone what your program does. This is called **commenting**. Comments are **not** part of a program and they do **not** affect how the program works.

Example

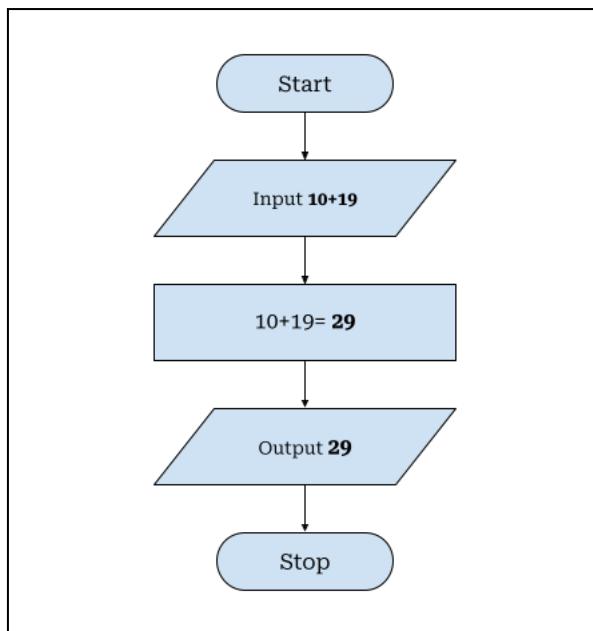
Use Sofya to calculate **10+19** and make a comment that says, "**This program calculates 10+19**".

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

```
Start this program
    Note: This program calculates 10+19;
    Write 10+19 on the screen
Stop this program
```

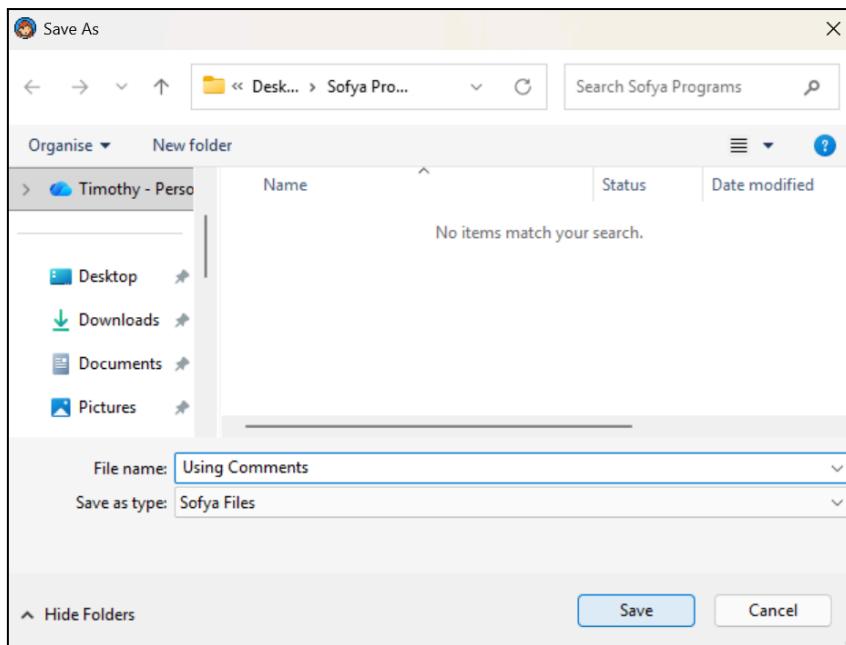
2. On the next page, there is a program flowchart for this program.



3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer that we want to start making a comment (when you want to start making a comment in Sofya, you should say "**Note:**"). Then we type the comment that we want. Then we tell the computer that we want to stop making a comment (when you want to stop making a comment in Sofya, you should use a semicolon [";"]).
- **Line 3:** We are telling the computer to write "**10+19**". When you tell Sofya to write an expression, the computer will calculate the expression first and then the computer will write the answer of the expression.
- **Line 4:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



5. The computer will write "29.0" on the screen.

```
>>> Program's Output <<<  
  
>>> THE PROGRAM HAS STARTED...  
  
29.0  
  
>>> THE PROGRAM HAS ENDED
```

Using Variables

A **variable** is something that **stores** things that keep on **changing**. Variables can be used for **calculations**.

Example

Use Sofya to calculate the **area** of a **circle** that has a radius of **8.8 cm** (the formula of finding the area of a circle is πr^2 . Use Pi[π] as **3.14**).

Solution

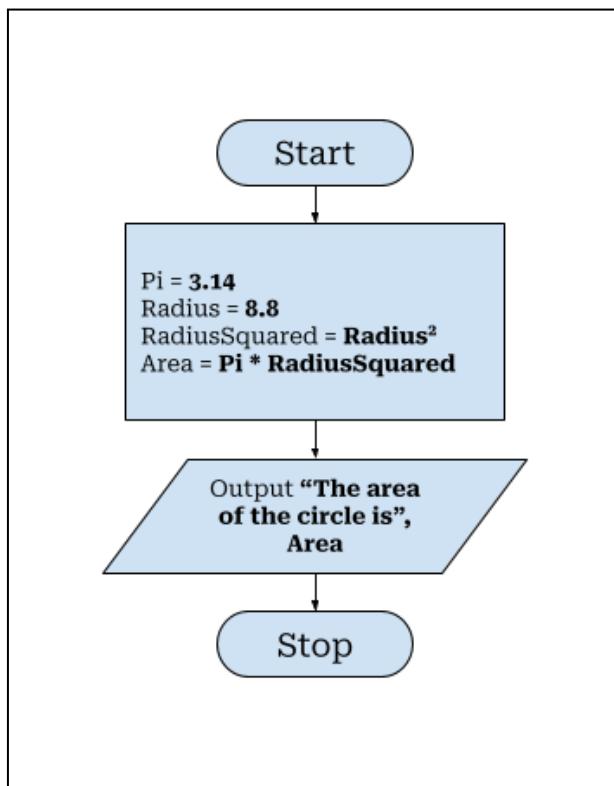
1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

- **Line 1:** Start this program

Have fun using Sofya Version 1.0!

- **Line 2:** Variable Pi is 3.14
- **Line 3:** Variable Radius is 8.8
- **Line 4:** Variable Area is Variable[Pi] x Variable[Radius] exp 2
- **Line 5:** Write "The area of the circle is" and also write Variable[Area] on the screen
- **Line 6:** Stop this program

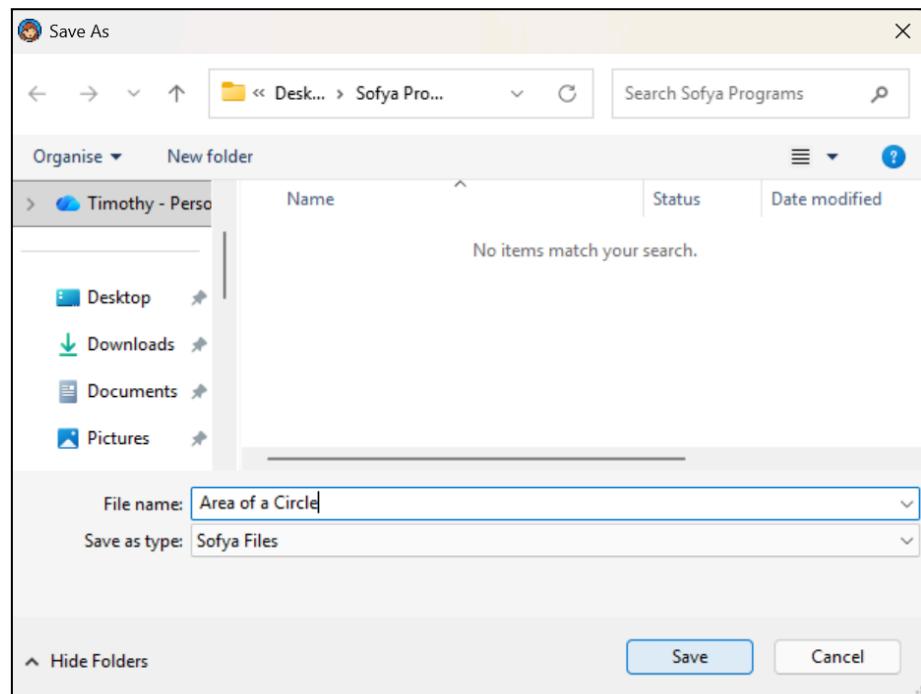
2. Here is a program flowchart for this program:



3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer that we are making a **variable** called **Pi** and that Pi is **3.14**.
- **Line 3:** We are telling the computer that we are making a **variable** called **Radius** and that Radius is **8.8**.
- **Line 4:** We are reminding the computer that **Radius** and **Pi** are **variables** and we want the computer to calculate **Pi x Radius exp 2** then store the answer in a **variable** called **Area**.

- **Line 5:** We are telling the computer to write "**The area of the circle is**" and what is stored in the variable called **Area**.
 - **Line 6:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").
4. Save this Sofya program and then run it.



5. The computer will write "**The area of the circle is**" and what is stored in the variable called **Area**.

```
>>> Program's Output <<<  
  
>>> THE PROGRAM HAS STARTED...  
  
THE AREA OF THE CIRCLE IS 243.16160000000005  
  
>>> THE PROGRAM HAS ENDED
```

Rules about Variables

1. The name of a variable should **not** have any **spaces** but it **can** have an **underscore** ("_"). For example, it is **not right** to call a variable

Radius Squared but it is **correct** to call a variable **RadiusSquared** or **Radius_Squared**.

2. A variable can be used to store a **string**, **number**, **expression**, **constant** or even another **variable**.
3. In Sofya, there are two ways of how variables can be used. Let us look at these two scenarios:

- **Scenario 1:** Variable Radius is 8.8
- **Scenario 2:** Write Variable[Radius] on the screen

In **scenario 1**, we are telling the computer that we are making a **new variable** called **Radius** and we are saying that Radius is **8.8**. In **scenario 2**, we are telling the computer to **get what was stored** in a variable called **Radius** and then we want the computer to write what was stored in "Radius" on the screen.

Receiving Input

Sometimes, you might want to ask the person who is using the computer some **questions** so that you can make a program. This is called **receiving input**.

Example

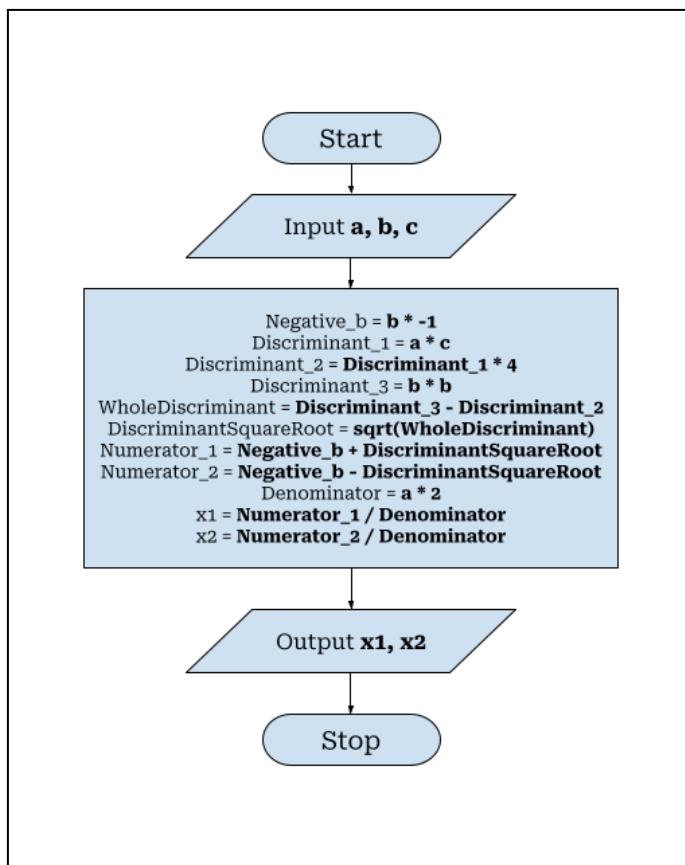
Use Sofya to **solve** the quadratic equation $2x^2 + x - 12 = 0$, using the formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ (This program will only work if $a \neq 0$ and if the **discriminant** $[b^2 - 4ac] \geq 0$).

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Ask The Computer User "Input the value of a" and store the answer in Variable[a]
 - **Line 3:** Ask The Computer User "Input the value of b" and store the answer in Variable[b]

- **Line 4:** Ask The Computer User "Input the value of c" and store the answer in Variable[c]
- **Line 5:** Variable x1 is $(-1 \times \text{Variable}[b] + (\text{Variable}[b]^2 - 4 \times \text{Variable}[a] \times \text{Variable}[c]) \exp(1/2)) / (2 \times \text{Variable}[a])$
- **Line 6:** Variable x2 is $(-1 \times \text{Variable}[b] - (\text{Variable}[b]^2 - 4 \times \text{Variable}[a] \times \text{Variable}[c]) \exp(1/2)) / (2 \times \text{Variable}[a])$
- **Line 7:** Write "x1 =" and also write Variable[x1] on the screen
- **Line 8:** Write "x2 =" and also write Variable[x2] on the screen
- **Line 9:** Stop this program

2. Here is a program flowchart for this program:

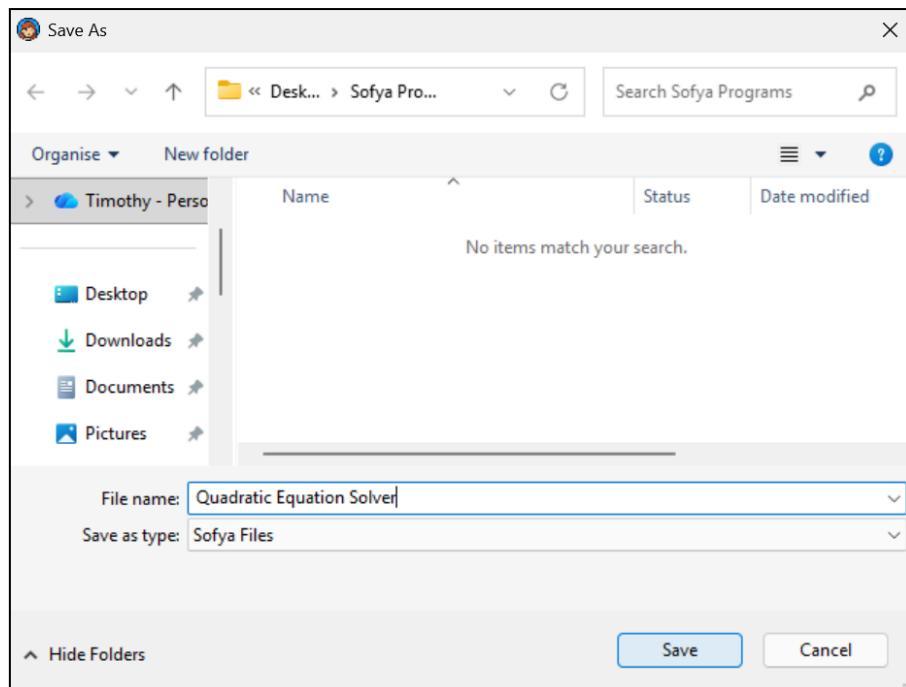


3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").

- **Line 2:** We are telling the computer to ask the person who is using the computer to input the **value** of "a" and after the person answers this question, we want the computer to store the person's answer in a **variable** called "a".
- **Line 3:** We are telling the computer to ask the person who is using the computer to input the **value** of "b" and after the person answers this question, we want the computer to store the person's answer in a **variable** called "b".
- **Line 4:** We are telling the computer to ask the person who is using the computer to input the **value** of "c" and after the person answers this question, we want the computer to store the person's answer in a **variable** called "c".
- **Line 5:** We are telling the computer that we are making a **variable** called **x1** and that it is $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$
- **Line 6:** We are telling the computer that we are making a **variable** called **x2** and that it is $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$
- **Line 7:** We are telling the computer to write "**x1** =" and what is stored in the variable called **x1**.
- **Line 8:** We are telling the computer to write "**x2** =" and what is stored in the variable called **x2**.
- **Line 9:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



- The computer will ask you to "Input the value of a". Type the number "2" in the input box and press "Enter" on the keyboard.

```
>>> Input Box <<<
2
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE VALUE OF A {Please answer this question in the input box}
```

- The computer will ask you to "Input the value of b". Type the number "1" in the input box and press "Enter" on the keyboard.

```
>>> Input Box <<<
1
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE VALUE OF A {Please answer this question in the input box}
> What you typed in the input box is: 2

INPUT THE VALUE OF B {Please answer this question in the input box}
```

- The computer will ask you to "Input the value of c". Type the number "-12" in the input box and press "Enter" on the keyboard.

```
>>> Input Box <<<
-12
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE VALUE OF A {Please answer this question in the input box}
> What you typed in the input box is: 2

INPUT THE VALUE OF B {Please answer this question in the input box}
> What you typed in the input box is: 1

INPUT THE VALUE OF C {Please answer this question in the input box}
```

8. If you scroll down in the program's output area, you will see that the computer will write "x1 =", "x2 =" and what is stored in the variables called **x1** and **x2**.

```
>>> Program's Output <<<

INPUT THE VALUE OF B {Please answer this question in the input box}
> What you typed in the input box is: 1

INPUT THE VALUE OF C {Please answer this question in the input box}
> What you typed in the input box is: -12

X1 = 2.212214450449026
X2 = -2.712214450449026

>>> THE PROGRAM HAS ENDED
```

Receiving Special Input

We have already seen how you can input **whole numbers (integers)** and **decimal numbers (floating point numbers)** in Sofya. Sometimes, you might want to input **special numbers**. In Sofya, the special numbers that you can input are: **numbers in standard (or scientific) form**, **improper fractions** and **mixed fractions**. This table below explains how you can input special numbers in Sofya.

If you want to input:	Then type it like this (in the input box) using the keyboard
A number in scientific form, for example, 1.024×10^{12}	$1.024 \times 10 \text{ exp } 12$

An improper fraction, for example, $\frac{22}{7}$	22/7
A mixed fraction, for example, $3\frac{1}{9}$	3 and 1/9

E-Form

Sometimes, the answer for a calculation can be a very **big number** or a very **small number**. If this happens, Sofya will write the answer in **e-form** (an example of a number written in e-form is **6.022e-23**). If you want to change a number from e-form to standard form, change the "e" to $\times 10^n$. The "n" is the **power**, which is on the **right side** of the "e". For example, the number **9.11e+14** in standard form will be **9.11 $\times 10^{+14}$** or **9.11 $\times 10^{14}$** .

If...Then Statements

Sometimes, you might want a computer to **compare** two things using a **condition** and if the condition is **true**, then the computer **will do** what you told it to do, but if the condition is **false** then the computer **will not do** what you told it to do. This is called an **If...Then statement**.

Comparison Operators

In Sofya, if you want to make a **condition**, you should use **comparison operators**. Here are the comparison operators that you can use in Sofya:

The name of the Comparison operator	How it looks in Sofya	The meaning of the operation	Example
Equal to	=	When you use the equal to operation, it means that a number is the same as another number.	2 = 2
Not Equal to	=/=	When you use the not equal to operation, it means that a number is not the same as another number.	10 =/= 7

Less than	<	When you use the less than operation, it means that a number is smaller than another number.	$3 < 5$
Greater than	>	When you use the greater than operation, it means that a number is bigger than another number.	$5 > 3$
Greater than or Equal to	> or =	When you use the greater than or equal to operation, it means that a number can be bigger than another number or that a number can be the same as another number. For example, number 5 is not the same as number 3 but number 5 is bigger than number 3.	$5 > \text{or } = 3$
Less than or Equal to	< or =	When you use the less than or equal to operation, it means that a number can be smaller than another number or that a number can be the same as another number. For example, number 7 is not the same as number 9 but number 7 is smaller than number 9.	$7 < \text{or } = 9$

Example

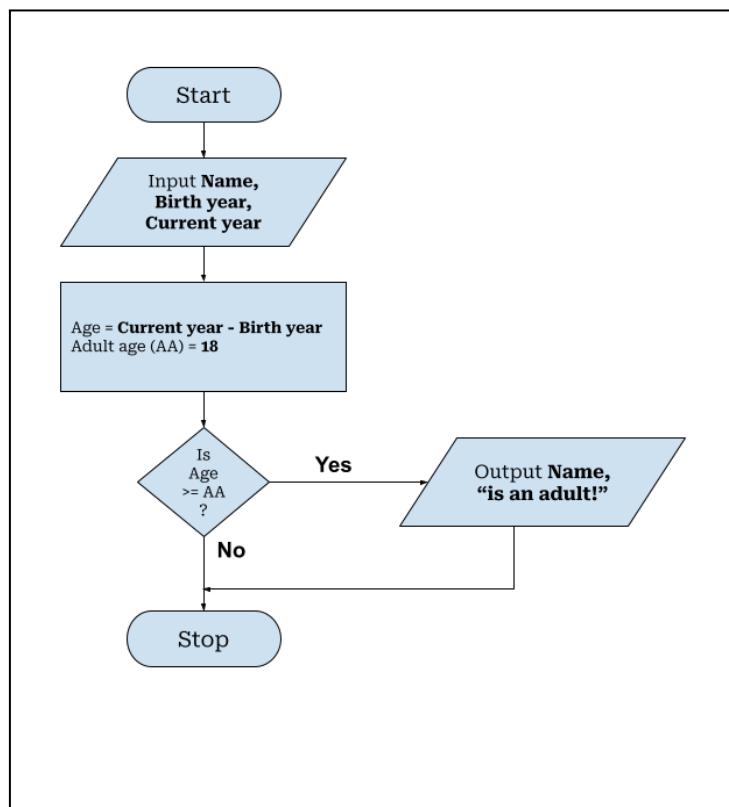
Alexia wanted to program her computer using Sofya, so that the computer could ask someone for their **name**, the **current year** and the **year when they were born** and then the computer could be able to know whether someone is an adult (An adult is 18 years and above). When the computer knows that someone is an adult, the computer will say the **name of the person** and, "*is an adult!*". **Angela** (Alexia's friend) was born in **2005**. Use Sofya to help Alexia program her computer (Assume that the current year is **2023**).

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:

- **Line 1:** Start this program
- **Line 2:** Ask The Computer User "What is your name?" and store the answer in Variable[Name]
- **Line 3:** Ask The Computer User "When were you born?" and store the answer in Variable[BirthYear]
- **Line 4:** Ask The Computer User "What is the current year?" and store the answer in Variable[CurrentYear]
- **Line 5:** Variable Age is Variable[CurrentYear] - Variable[BirthYear]
- **Line 6:** If Variable[Age] > or = 18 Then
- **Line 7:** Write Variable[Name] and also write "is an adult!" on the screen
- **Line 8:** End this If Statement
- **Line 9:** Stop this program

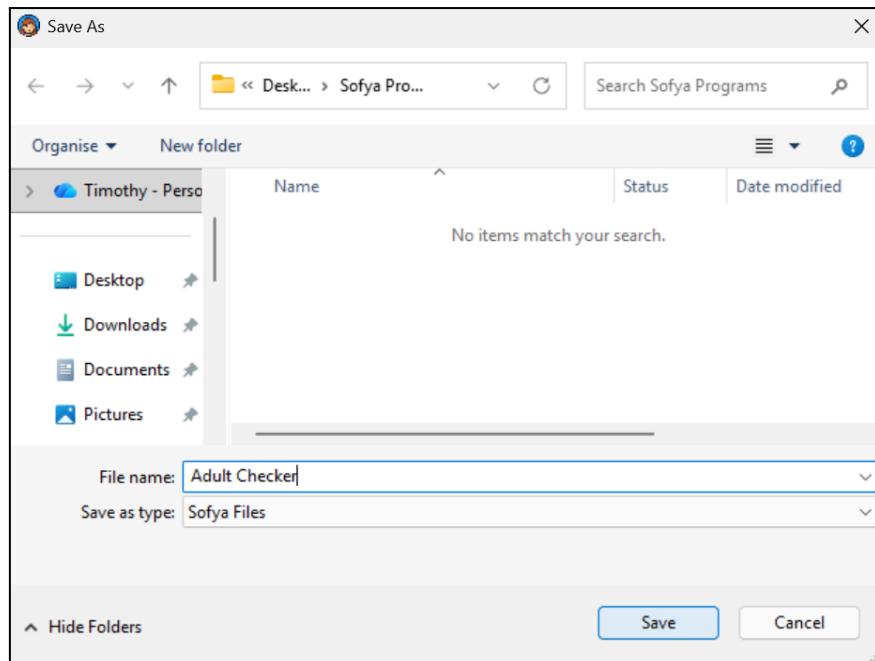
2. Here is a program flowchart for this program:



3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to ask the person who is using it, "**What is your name?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Name**.
- **Line 3:** We are telling the computer to ask the person who is using it, "**When were you born?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **BirthYear**.
- **Line 4:** We are telling the computer to ask the person who is using it, "**What is the current year?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **CurrentYear**.
- **Line 5:** We are telling the computer that we are making a **variable** called **Age** and that **Age** is **CurrentYear - BirthYear**.
- **Line 6:** We are telling the computer to check if the **Age** is greater than or equal to **18**.
- **Line 7:** We are telling the computer to write what is stored in the **variable** called **Name** and also write the words "**is an adult!**" (This will only happen if the **age** is greater than or equal to **18**).
- **Line 8:** We are telling the computer that we want to stop making the If...Then statement (That is why we put "**End this If Statement**").
- **Line 9:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



- The computer will ask you, "**What is your name?**". Type "**Angela**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
Angela
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR NAME? {Please answer this question in the input box}
```

- The computer will ask you, "**When were you born?**". Type "**2005**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
2005
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR NAME? {Please answer this question in the input box}
> What you typed in the input box is: Angela

WHEN WERE YOU BORN? {Please answer this question in the input box}
```

- The computer will ask you, "**What is the current year?**". Type "**2023**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
2023
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR NAME? {Please answer this question in the input box}
> What you typed in the input box is: Angela

WHEN WERE YOU BORN? {Please answer this question in the input box}
> What you typed in the input box is: 2005

WHAT IS THE CURRENT YEAR? {Please answer this question in the input box}
```

8. If you scroll down the program's output, you will see that the computer will write what is stored in the variable called **Name** and "**is an adult!**".

```
>>> Program's Output <<<
> What you typed in the input box is: Angela

WHEN WERE YOU BORN? {Please answer this question in the input box}
> What you typed in the input box is: 2005

WHAT IS THE CURRENT YEAR? {Please answer this question in the input box}
> What you typed in the input box is: 2023

ANGELA IS AN ADULT!

>>> THE PROGRAM HAS ENDED
```

Rules about If...Then Statements

1. The two things that you are comparing in the condition can be:
 - A **variable first** then another **variable**, for example,
If Variable [x] = Variable [b] Then
 - A **variable first** then a **number**, for example,
If Variable [x] = 24 Then
 - A **variable first** then an **expression**, for example,
If Variable [x] = 22/7 Then
 - A **variable first** then a **string**, for example,
If Variable [x] = "Sofya" Then
2. If a condition is **false**, the computer will not do what you told it to do in the If...Then statement.

3. At the **end** of an If...Then statement, you should **always** put "End this If Statement" the way we have done in this example.
4. You **can** make If...Then statements, If...Then...Else statements or If...Otherwise-If...Else Statements inside of other If...Then statements.
5. You **can** make **Do This...Until Loops** inside of If...Then statements.

Constants

Constants are things that do not change, for example, the speed of light does not change. In Sofya, you can use up to **3 mathematical constants** and **15 scientific constants** for your calculations. This table below shows the list of constants that are in Sofya Version 1.0.

The name of the constant	How it looks in Sofya	The value of the constant
Pi (π)	Constant[Pi]	3.14159265358979323846
Speed of light in a vacuum (c)	Constant[c]	299792458 ms ⁻¹
Planck's constant (h)	Constant[h]	$6.62607015 \times 10^{-34}$ Js
Permittivity of free space (ϵ_0)	Constant[E_0] or Constant[E_0]	$8.854187812813 \times 10^{-12}$ Fm ⁻¹
Newton's constant of gravitation (G)	Constant[G]	$6.6743015 \times 10^{-11}$ m ³ kg ⁻¹ s ⁻²
Coulomb constant (k_e)	Constant[k_e]	$8.987551792314 \times 10^9$ Nm ² C ⁻²
Acceleration of gravity on Earth (g_e)	Constant[g_e]	9.80665 ms ⁻²
Earth's atmospheric pressure at sea level (1 atm)	Constant[atm]	101,305 Pa
Mass of an electron (m_e)	Constant[m_e]	$9.109383701528 \times 10^{-31}$ kg
Mass of a proton (m_p)	Constant[m_p]	$1.6726219236951 \times 10^{-27}$ kg
Mass of a neutron (m_n)	Constant[m_n]	$1.6749274980495 \times 10^{-27}$ kg

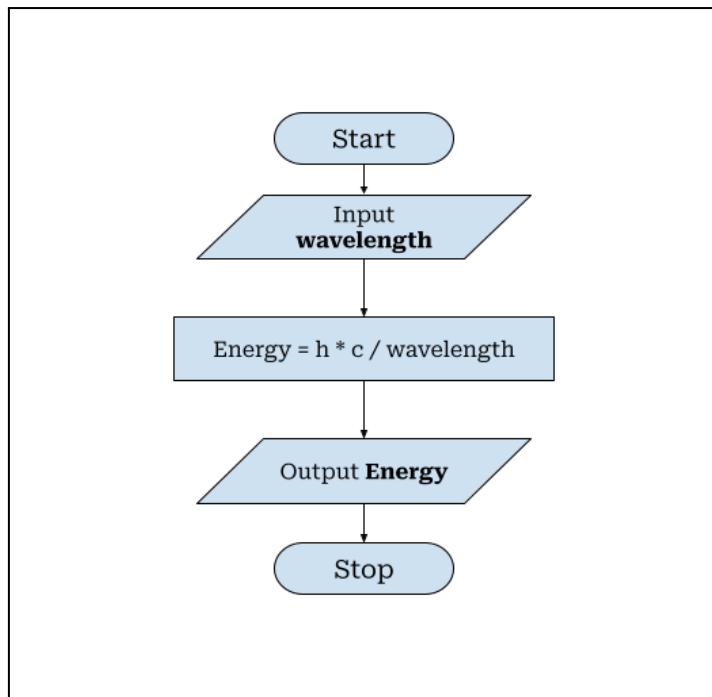
Proton to electron mass ratio ($\mathbf{m_p/m_e}$)	Constant[m_p:m_e]	1,836.1526734311
Avogadro's Number ($\mathbf{N_a}$ or \mathbf{L})	Constant[N_a] or Constant[L]	$6.02214076 \times 10^{23}$ substances mol $^{-1}$
Molar gas constant (\mathbf{R})	Constant[R]	8.31446261815324 Jmol $^{-1}$ K $^{-1}$
Faraday constant (\mathbf{F})	Constant[F]	96,485.3321233100184 Cmol $^{-1}$
Radius of an electron ($\mathbf{r_e}$)	Constant[r_e]	$2.817940326213 \times 10^{-15}$ m
Golden ratio (Φ)	Constant[Phi]	1.61803398874989484820
Euler's number (e)	Constant[e]	2.71828182845904523536

Example

Green light has a wavelength of 5.17×10^{-7} m. Using the formula, $E = h\frac{c}{\lambda}$ (where E is the *energy of light*, h is *Planck's constant*, c is the *speed of light* and λ is the *wavelength of light*), use Sofya to create a program to calculate the energy the green light has.

Solution

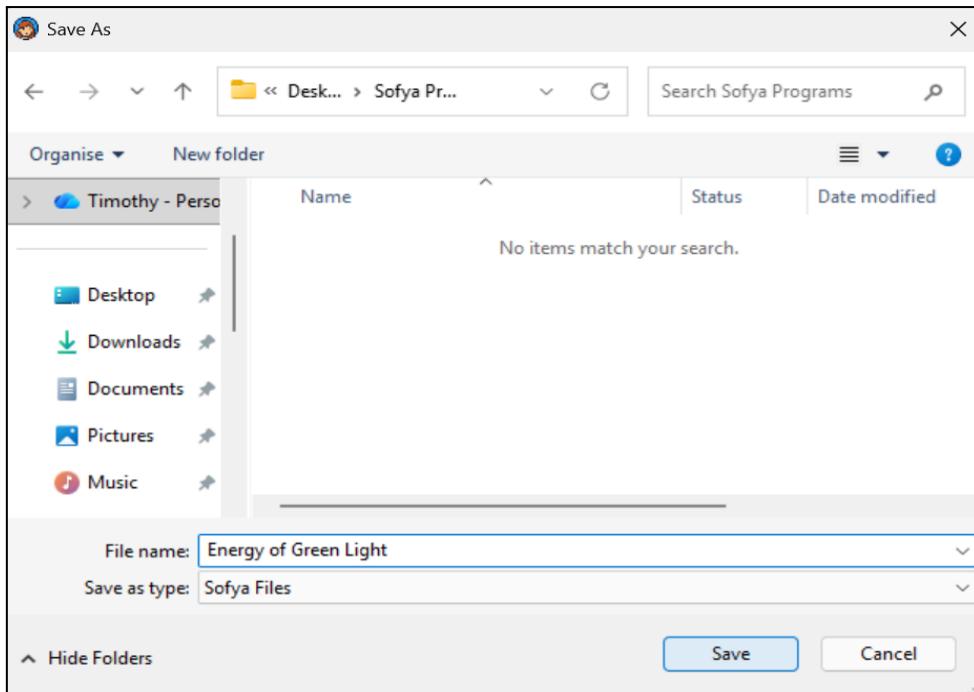
1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Ask The Computer User "What is the wavelength of the green light?" and store the answer in Variable[Wavelength]
 - **Line 3:** Variable Energy is Constant[h] x Constant[c] / Variable[Wavelength]
 - **Line 4:** Write "The energy of the green light is" and also write Variable[Energy] and also write "Joules" on the screen
 - **Line 5:** Stop this program
2. Here is a program flowchart for this program:



3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to ask the person who is using it, "**What is the wavelength of the green light?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Wavelength**.
- **Line 3:** We are telling the computer that we are making a **variable** called **Energy** and that Energy is $h \frac{c}{\lambda}$.
- **Line 4:** We are telling the computer to write what is stored in the variable called **Energy** and the words "**The energy of the green light is**" and the word "**Joules**".
- **Line 5:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



- The computer will ask you, "**What is the wavelength of the green light?**". Type "**5.17 x 10 exp -7**" in the input box and press "Enter" on the keyboard.

```
>>> Input Box <<<
5.17 x 10 exp -7
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS THE WAVELENGTH OF THE GREEN LIGHT? {Please answer this question in the input box}
```

- The computer will write what is stored in the variable called **Energy** and the words "**The energy of the green light is**" and the word "**Joules**".

```
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS THE WAVELENGTH OF THE GREEN LIGHT? {Please answer this question in the input box}
> What you typed in the input box is: 5.17 x 10 exp -7

THE ENERGY OF THE GREEN LIGHT IS 3.842255042841255e-19 JOULES

>>> THE PROGRAM HAS ENDED
```

If...Then...Else Statements

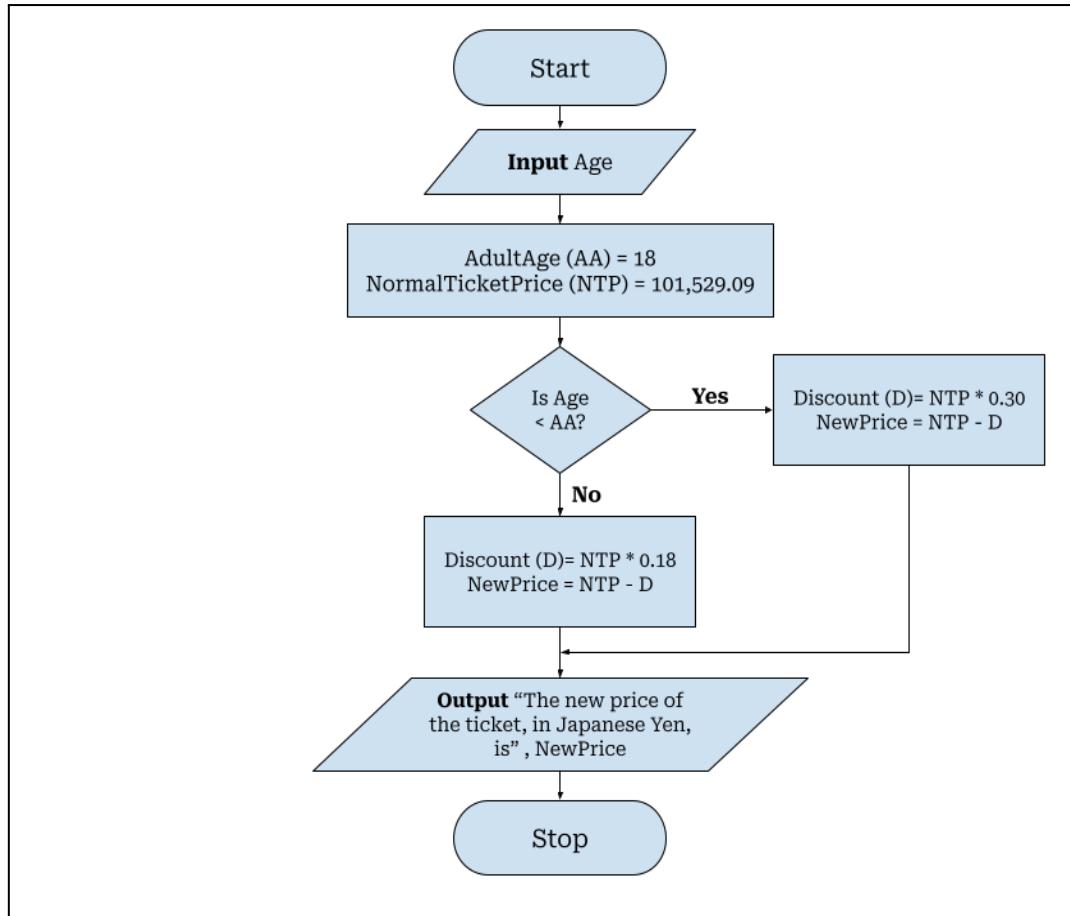
Sometimes, you might want a computer to **compare** two things using a **condition** and if the condition is **true**, then the computer **will do** what you told it to do, but if the condition is **false** then you can tell the computer what to do instead. This is called an **If...Then...Else statement**.

Example

Amai Airways is an **airline company** in Japan. On **New Year's Day**, they give discounts on aeroplane tickets depending on a person's age. If a person is 18 years and above (an adult), they get a discount of **18%** on their aeroplane ticket. However, if a person is below 18 years, they get a discount of **30%** on their aeroplane ticket. If, without a discount, an aeroplane ticket costs **101,529.09¥**, use Sofya to create a program to automatically calculate the new price of an aeroplane ticket after someone is given a discount.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Ask The Computer User "What is your age?" and store the answer in Variable[Age]
 - **Line 3:** Variable NormalTicketPrice is 101529.09
 - **Line 4:** If Variable[Age] > or = 18 Then
 - **Line 5:** Variable Discount is
Variable[NormalTicketPrice] x 0.18
 - **Line 6:** Variable NewPrice is
Variable[NormalTicketPrice] - Variable[Discount]
 - **Line 7:** Otherwise If Variable[Age] is anything else Then
 - **Line 8:** Variable Discount is
Variable[NormalTicketPrice] x 0.30
 - **Line 9:** Variable NewPrice is
Variable[NormalTicketPrice] - Variable[Discount]
 - **Line 10:** End this If Statement
 - **Line 11:** Write "The new price of the ticket is" and also write Variable[NewPrice] and also write "Japanese Yen" on the screen
 - **Line 12:** Stop this program
2. On the next page there is a program flowchart for this program.

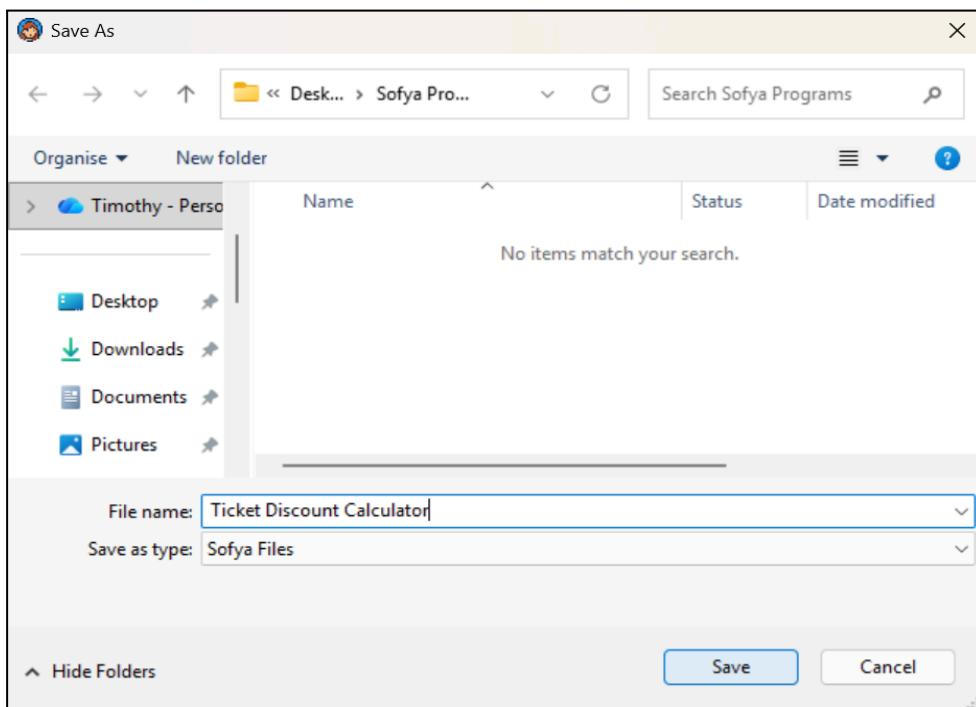


3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to ask the person who is using it, "**What is your age?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Age**.
- **Line 3:** We are telling the computer that we are making a **variable** called **NormalTicketPrice** and that NormalTicketPrice is **101,529.09**.
- **Line 4:** We are telling the computer to check if the **age** is greater than or equal to 18.
- **Line 5:** We are telling the computer that we are making a **variable** called **Discount** and that Discount is **NormalTicketPrice x 0.18** (This will only happen if the **Age** is greater than or equal to the **18**).

- **Line 6:** We are telling the computer that we are making a **variable** called **NewPrice** and that NewPrice is **NormalTicketPrice - Discount** (This will only happen if the **Age** is greater than or equal to the **18**).
- **Line 7:** We are telling the computer to check if the **Age** is any other number (which is not greater than or equal to 18).
- **Line 8:** We are telling the computer that we are making a **variable** called **Discount** and that Discount is **NormalTicketPrice x 0.30** (This will only happen if the **Age** is less than 18).
- **Line 9:** We are telling the computer that we are making a **variable** called **NewPrice** and that NewPrice is **NormalTicketPrice - Discount** (This will only happen if the **Age** is less than 18).
- **Line 10:** We are telling the computer that we want to stop making the If...Then..Else statement (That is why we put "**End this If Statement**").
- **Line 11:** We are telling the computer to write what is stored in the variable called **NewPrice** and the words "**The new price of the ticket is**" and the words "**Japanese Yen**".
- **Line 12:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



5. The computer will ask you, "**What is your age?**". Type "**10**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
10
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR AGE? {Please answer this question in the input box}
```

6. The computer will write what is stored in the variable called **NewPrice** and the words "**The new price of the ticket is**" and the words "**Japanese Yen**".

```
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR AGE? {Please answer this question in the input box}
> What you typed in the input box is: 10

THE NEW PRICE OF THE TICKET IS 71,070.363 JAPANESE YEN

>>> THE PROGRAM HAS ENDED
```

Rules about If...Then...Else Statements

1. The two things that you are comparing in the condition can be:
 - A **variable first** then another **variable**, for example,
If Variable [x] = Variable [b] Then
 - A **variable first** then a **number**, for example,
If Variable [x] = 24 Then
 - A **variable first** then an **expression**, for example,
If Variable [x] = 22/7 Then
 - A **variable first** then a **string**, for example,
If Variable [x] = "Sofya" Then
2. At the **end** of an If...Then...Else statement, you should **always** put "**End this If Statement**" the way we have done in this example.

3. You **can** make If...Then statements, If...Then...Else statements or If...Otherwise-If...Else Statements inside of other If...Then...Else statements.
4. You **can** make **Do This...Until Loops** inside of If...Then...Else statements.

Using Ranges

A range is a group of numbers. Let us say that we have items that have a price range of **\$10** to **\$50**. If we say that:

- *The price of 5kgs of sugar, for example, is from \$10 to \$50*, it means that *the sugar can cost \$10 or more but the price of sugar can only go up to \$50 but it cannot cost more than \$50* (So, we can say that **\$10 and \$50 are included** in the range). If we wanted to make a condition for this in Sofya, it can look like this: **If Variable[SugarPrice] is From 10 to 50 Then.**
- *The price of 5kgs of sugar, for example, is from after \$10 to \$50*, it means that *the sugar can cost more than \$10 but the price of sugar can only go up to \$50 but it cannot cost more than \$50* (So, we can say that **\$10 is not included** in the range but **\$50 is included** in the range). If we wanted to make a condition for this in Sofya, it can look like this: **If Variable[SugarPrice] is From After 10 to 50 Then.**
- *The price of 5kgs of sugar, for example, is from \$10 to before \$50*, it means that *the sugar can cost \$10 or more but the price of sugar cannot reach \$50 and above* (So, we can say that **\$10 is included** in the range but **\$50 is not included** in the range). If we wanted to make a condition for this in Sofya, it can look like this: **If Variable[SugarPrice] is From 10 to Before 50 Then.**
- *The price of 5kgs of sugar, for example, is between \$10 and \$50*, it means that *the sugar can cost all the prices which are more than \$10 but less than \$50* (So, we can say that **\$10 and \$50 are not included** in the range). If we wanted to make a condition for this in Sofya, it can look like this: **If Variable[SugarPrice] is Between 10 and 50 Then.**

Rules of Ranges

1. At the **start** or the **end** of a range you can use a **number** or **expression**. For example, it is correct to say:

If Variable[x] is from 3 to 12 Then

Or

If Variable[x] is from 1/3 to 3 Then

Or

If Variable[x] is from 5 to 31/3 Then

Or

If Variable[x] is from 1/6 to 100/3 Then

2. The number or expression at the **start** of a range should always be **smaller** than the number or expression used at the **end** of a range. For example, it is correct to say, **If Variable[x] is from 1 to 26 Then**, but it is wrong to say, **If Variable[x] is from 26 to 1 Then**.

If...Otherwise-If...Else Statements

Sometimes, you might want a computer to check **many conditions (more than two)** conditions and then the computer would only do the condition that is **true**. This is called a **If...Otherwise-If...Else Statement** or a **Multiple If Statement**.

Example

In a computer game, players are ranked based on their points as shown in the table below (points in computer games are also called **experience** or **XP**):

XP	Rank of the Player
0	Beginner or Noob
Above 0 to 10,000	1 st Stage Intermediate
Above 10,000 to 100,000	2 nd Stage Intermediate
Above 100,000 to 1,000,000	Pro
Above 1,000,000 to 10,000,000	Expert
Above 10,000,000 to 20,000,000	Master
Above 20,000,000 to 50,000,000	Extraordinary Gamer
Above 50,000,000	Gaming Legend

In this computer game, XP is an **integer** (in this game, XP cannot be a decimal or floating point number). Also, in this game if a player is found cheating (hacking the game to get an advantage), their XP will be a **negative number** (below 0 XP) and their rank will be '**Cheater**'. Use Sofya to create a program to rank players using their XP.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Ask The Computer User "What is your username in the game?" and store the answer in Variable[Username]
 - **Line 3:** Ask The Computer User "What is your XP in the game?" and store the answer in Variable[xp]
 - **Line 4:** If Variable[xp] = 0 Then
 - **Line 5:** Write Variable[Username] and also write "is a Noob" on the screen
 - **Line 6:** Otherwise if Variable[xp] is from 1 to 10000 Then
 - **Line 7:** Write Variable[Username] and also write "is a 1st Stage Intermediate" on the screen
 - **Line 8:** Otherwise if Variable[xp] is from 10001 to 100000 Then
 - **Line 9:** Write Variable[Username] and also write "is a 2nd Stage Intermediate" on the screen
 - **Line 10:** Otherwise if Variable[xp] is from 100001 to 1000000 Then
 - **Line 11:** Write Variable[Username] and also write "is a Pro" on the screen
 - **Line 12:** Otherwise if Variable[xp] is from 1000001 to 10000000 Then
 - **Line 13:** Write Variable[Username] and also write "is an Expert" on the screen
 - **Line 14:** Otherwise if Variable[xp] is from 10000001 to 20000000 Then

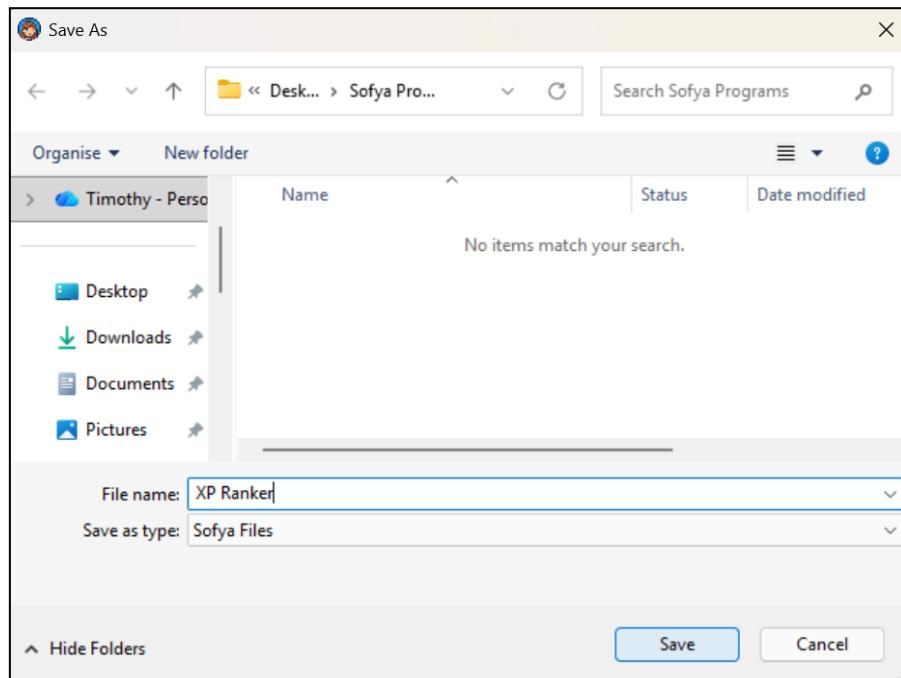
- **Line 15:** Write Variable[Username] and also write "is a Master" on the screen
- **Line 16:** Otherwise if Variable[xp] is from 20000001 to 50000000 Then
- **Line 17:** Write Variable[Username] and also write "is an Extraordinary Gamer" on the screen
- **Line 18:** Otherwise if Variable[xp] > 50000000 Then
- **Line 19:** Write Variable[Username] and also write "is a Gaming Legend" on the screen
- **Line 20:** Otherwise if Variable[xp] is anything else Then
- **Line 21:** Write Variable[Username] and also write "is a Cheater" on the screen
- **Line 22:** End this If Statement
- **Line 23:** Stop this program

2. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer to ask the person who is using it, "**What is your username in the game?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Username**.
- **Line 3:** We are telling the computer to ask the person who is using it, "**What is your XP in the game?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **xp**.
- **Line 4 - 21:** We are telling the computer to check the **XP** of a player and the computer will rank a player using their XP.
- **Line 22:** We are telling the computer that we want to stop making the If...Otherwise-If...Else Statement (That is why we put "**End this If Statement**").

- **Line 23:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

3. Save this Sofya program and then run it.



4. The computer will ask you, "**What is your username in the game?**". You can type "**Bok**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
Bok
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR USERNAME IN THE GAME? {Please answer this question in the input box}
```

5. The computer will ask you, "**What is your XP in the game?**". You can type "**80,000,000**" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
80000000
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR USERNAME IN THE GAME? {Please answer this question in the input box}
> What you typed in the input box is: Bok

WHAT IS YOUR XP IN THE GAME? {Please answer this question in the input box}
```

6. The computer will write what is stored in the variable called **Username** and the computer will show you your rank.

```
>>> Program's Output <<<
>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR USERNAME IN THE GAME? {Please answer this question in the input box}
> What you typed in the input box is: Bok

WHAT IS YOUR XP IN THE GAME? {Please answer this question in the input box}
> What you typed in the input box is: 80000000

BOK IS A GAMING LEGEND

>>> THE PROGRAM HAS ENDED
```

Rules about If...Otherwise-If...Else Statements

1. The two things that you are comparing in the condition can be:
 - A **variable first** then another **variable**, for example,
If Variable [x] = Variable [b] Then
 - A **variable first** then a **number**, for example,
If Variable [x] = 24 Then
 - A **variable first** then an **expression**, for example,
If Variable [x] = 22/7 Then
 - A **variable first** then a **string**, for example,
If Variable [x] = "Sofya" Then
2. At the **end** of a If...Otherwise-If...Else Statement, you should **always** put "**End this If Statement**" the way we have done in this example.
3. You **can** make If...Then statements, If...Then...Else statements or If...Otherwise-If...Else Statements inside of other If...Otherwise-If...Else Statements.
4. You **can** make **Do This...Until Loops** inside of If...Otherwise-If...Else Statements.

Nested If Statements

If...Then statements, *If...Then...Else* statements and *If...Otherwise-If...Else* statements are all called **if statements**. Sometimes, you might want to put an if statement **inside** of another if statement. When you do this, we say that you are making a **nested if statement**.

Example

Make a Sofya program that asks the computer user to input 3 **numbers** and the computer should tell you the **biggest** number among the 3 numbers.

Solution

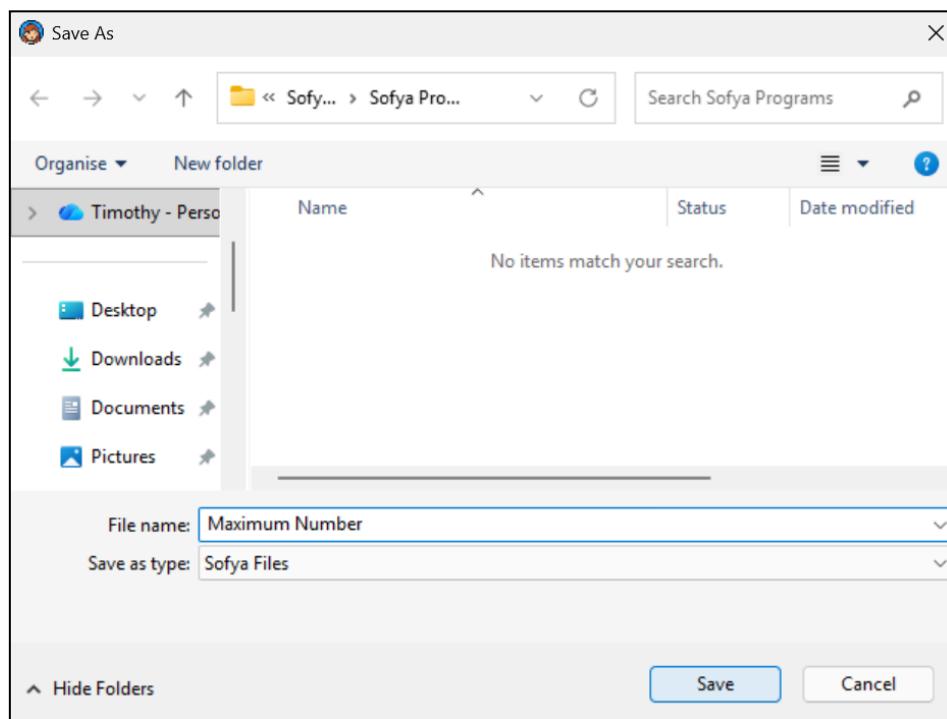
1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Ask The Computer User "Input the 1st number" and store the answer in Variable[1stNumber]
 - **Line 3:** Ask The Computer User "Input the 2nd number" and store the answer in Variable[2ndNumber]
 - **Line 4:** Ask The Computer User "Input the 3rd number" and store the answer in Variable[3rdNumber]
 - **Line 5:** If Variable[1stNumber] > Variable[2ndNumber]
Then
 - **Line 6:** Make this nested if statement
 - **Line 7:** {
 - **Line 8:** If Variable[1stNumber] > Variable[3rdNumber]
Then
 - **Line 9:** Write "The biggest number is" and also write Variable[1stNumber] on the screen
 - **Line 10:** End this if statement
 - **Line 11:** }
 - **Line 12:** Otherwise if Variable[2ndNumber] >
Variable[1stNumber] Then
 - **Line 13:** Make this nested if statement
 - **Line 14:** {

- **Line 15:** If Variable[2ndNumber] > Variable[3rdNumber]
Then
- **Line 16:** Write "The biggest number is" and also write Variable[2ndNumber] on the screen
- **Line 17:** End this if statement
- **Line 18:** }
- **Line 19:** Otherwise if Variable[3rdNumber] is anything else Then
- **Line 20:** Write "The biggest number is" and also write Variable[3rdNumber] on the screen
- **Line 21:** End this if statement
- **Line 22:** Stop this program

2. Here is an explanation of the new concepts:

- **Line 6 and Line 7:** We are telling the computer that we are making a nested if statement (We are using an opening curly brace ["{ "] to tell the computer where the nested if statement starts).
- **Line 11:** We are using a closing curly brace ["}"] to tell the computer where the nested if statement ends.

3. Save this Sofya program and then run it.



4. The computer will ask you, “**Input the 1st number**”. Type “7,402” in the input box and press **Enter** on the keyboard.

```
>>> Input Box <<<
7402
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE 1ST NUMBER {Please answer this question in the input box}
```

5. The computer will ask you, “**Input the 2nd number**”. Type “6,829” in the input box and press **Enter** on the keyboard.

```
>>> Input Box <<<
6829
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE 1ST NUMBER {Please answer this question in the input box}
> What you typed in the input box is: 7402

INPUT THE 2ND NUMBER {Please answer this question in the input box}
```

6. The computer will ask you, “**Input the 3rd number**”. Type “6,937” in the input box and press **Enter** on the keyboard.

```
>>> Input Box <<<
6937
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

INPUT THE 1ST NUMBER {Please answer this question in the input box}
> What you typed in the input box is: 7402

INPUT THE 2ND NUMBER {Please answer this question in the input box}
> What you typed in the input box is: 6829

INPUT THE 3RD NUMBER {Please answer this question in the input box}
```

7. If you scroll down the program’s output, you will see that the computer will show you the biggest number on the screen.

```
>>> Program's Output <<<
> What you typed in the input box is: 7402

INPUT THE 2ND NUMBER {Please answer this question in the input box}
> What you typed in the input box is: 6829

INPUT THE 3RD NUMBER {Please answer this question in the input box}
> What you typed in the input box is: 6937

THE BIGGEST NUMBER IS 7,402.0

>>> THE PROGRAM HAS ENDED
```

Rules of Nested If Statements

1. You **can** make a nested if statement inside of **another** nested if statement.
2. Always remember to say "**Make this nested if statement**" and then you should use curly braces ({ }) to tell the computer that you want to make a nested if statement.

Static Variable Operations (SVOs)

Let us say that we want to make a variable called **Num**, so we say, Variable Num is 10. Let us also say that you want the computer to get what is stored in the variable called Num (in this example, we want the computer to get **10**). After the computer has gotten the number 10, we want the computer to do a **calculation** using 10 (for example, **10 + 5**) and after the computer does this, we want the computer to store the final answer in the **same variable** called **Num**. Since **Num** is the same variable where we were getting number **10** to do a calculation and then we are still storing the answer for this calculation in the variable called Num, we can say that **Num** is a **static variable**. When you use a static variable to do a calculation, we say that you are doing a **Static Variable Operation (SVO)**. If we wanted to do this SVO (that is talked about in this paragraph, in Sofya) we would say, Increase Variable [Num] by 5.

Any SVO (for example, Increase Variable [Num] by 5) has **4 parts**:

- A **Static Variable Operation Keyword**, for example, "Increase" (Static Variable Operation Keywords tell the computer which SVO to do).
- A **Static Variable**, for example, "Variable[Num]".

- A **Value Assigner**, for example, "by".
- A **Value Changer**, for example, "5".

Note: The **value changer** of a SVO **can** be:

- A **Number**, for example, Increase Variable [Num] by 5.
- An **Expression**, for example, Increase Variable [Num] by 1/7.
- A **Variable**, for example, Increase Variable [Num] by Variable [x].

Here are the **seven** Static Variable Operations (SVOs) that you can do in Sofya:

- **Static Variable Addition (SVA)** → This is when you want the computer to use a static variable to do **addition**.
- **Static Variable Subtraction (SVS)** → This is when you want the computer to use a static variable to do **subtraction**.
- **Static Variable Product (SVP)** → This is when you want the computer to use a static variable to do **multiplication**.
- **Static Variable Division (SVD)** → This is when you want the computer to use a static variable to do **division**.
- **Static Variable Exponentiation (SVE)** → This is when you want the computer to use a static variable to do **exponentiation**.
- **Static Variable Flooring (SVF)** → This is when you want the computer to use a static variable to do **flooring**.
- **Static Variable Modulus (SVM)** → This is when you want the computer to use a static variable to do **modulus**.

Here is a table to show you how to **use** all the different **SVOs** in Sofya (In this table, we are assuming that **Variable[x]** is **10**):

If you want to do this SVO:	Then the Static Variable Operation Keyword(s) is:	Then the Value Assigner is:	Example
Static Variable Addition	Increase	By	<p><i>Increase Variable[x] by 10</i></p> <p>Note: After doing this SVO, Variable[x] is 20</p>

<i>Static Variable Subtraction</i>	<i>Decrease</i>	<i>By</i>	<i>Decrease Variable[x] by 5</i> Note: After doing this SVO, Variable[x] is 5
<i>Static Variable Product</i>	<i>Multiply</i>		<i>Multiply Variable[x] by 15</i> Note: After doing this SVO, Variable[x] is 150
<i>Static Variable Division</i>	<i>Divide</i>		<i>Divide Variable[x] by 3</i> Note: After doing this SVO, Variable[x] is 3 1/3
<i>Static Variable Exponentiation</i>	<i>Do Exp For</i>	<i>Using</i>	<i>Do Exp for Variable[x] using 4</i> Note: After doing this SVO, Variable[x] is 10,000
<i>Static Variable Flooring</i>	<i>Do Flooring For</i>		<i>Do Flooring for Variable[x] using 3</i> Note: After doing this SVO, Variable[x] is 3
<i>Static Variable Modulus</i>	<i>Do Modulus For</i>		<i>Do Modulus for Variable[x] using 3</i> Note: After doing this SVO, Variable[x] is 1

Note: SVOs are **mostly** used in **Do This...Until loops**.

Do This...Until Loops

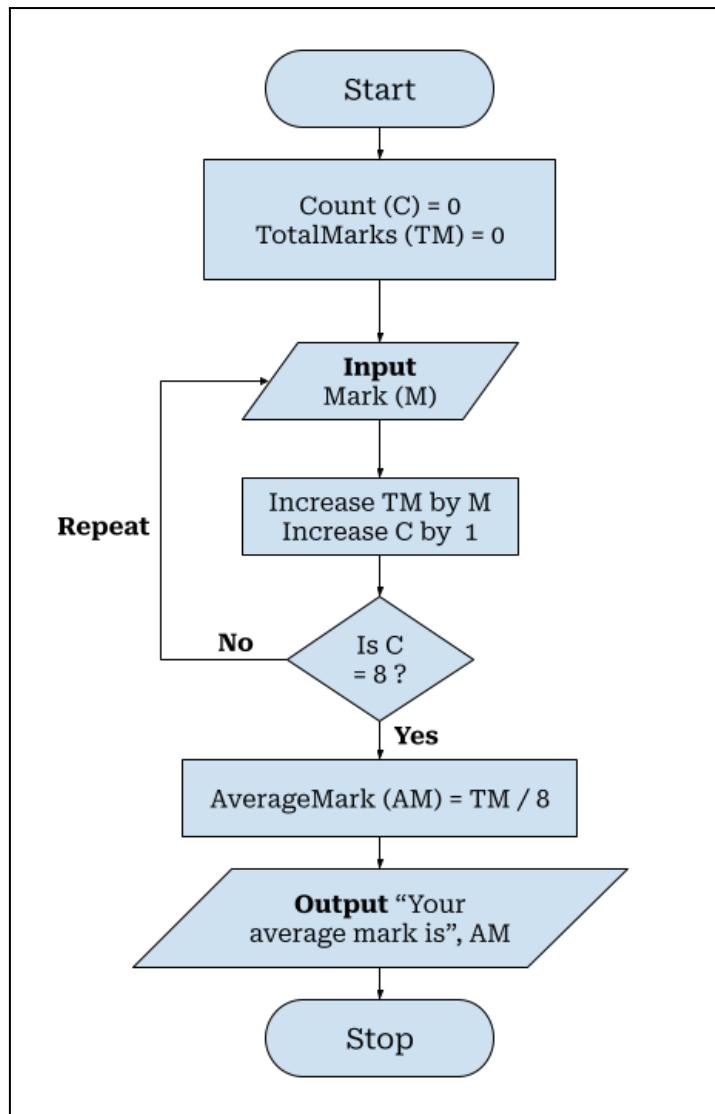
Sometimes, you might want a computer to **repeat** things using a **condition** and if the condition is **false**, then the computer **will continue** repeating what you told it to do until the condition becomes **true**. This is called a **Do This...Until Loop**. The things that you want the computer to repeat in the Do This...Until loop are called the **loop body**.

Example

Pacifica is a high school student. She did 8 examinations and scored the following marks (out of 100 marks): 50, 49, 67, 89, 92, 41, 73 and 65. Use Sofya to program a computer so that the computer would calculate her average mark.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Variable Count is 0
 - **Line 3:** Variable TotalMarks is 0
 - **Line 4:** Do This
 - **Line 5:** {
 - **Line 6:** Ask The Computer User "What mark did you score in the examination?" and store the answer in Variable[Mark]
 - **Line 7:** Increase Variable[TotalMarks] by Variable[Mark]
 - **Line 8:** Increase Variable[Count] by 1
 - **Line 9:** }
 - **Line 10:** Until Variable[Count] = 8
 - **Line 11:** Variable AverageMark is Variable[TotalMarks]/8
 - **Line 12:** Write "Your average mark is" and also write Variable[AverageMark] on the screen
 - **Line 13:** Stop this program
2. On the next page there is a program flowchart for this program.

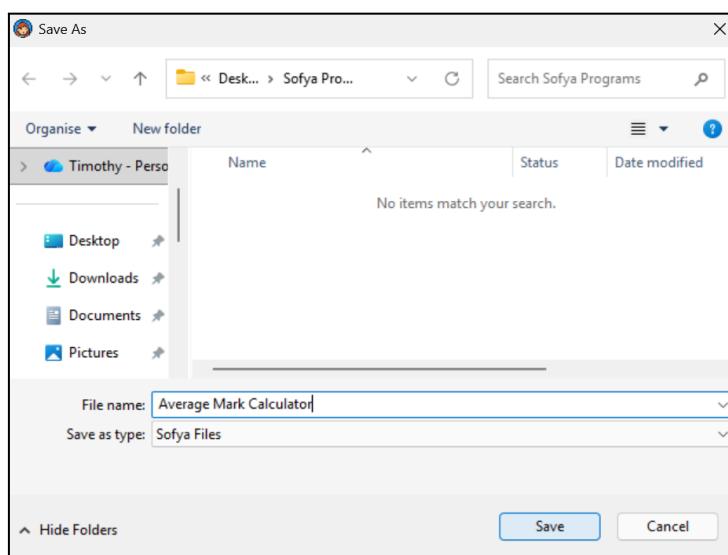


3. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer that we are making a **variable** called **Count** and that Count is **0**.
- **Line 3:** We are telling the computer that we are making a **variable** called **TotalMarks** and that TotalMarks is **0**.
- **Line 4 and Line 5:** We are telling the computer that we are making a Do This...Until loop (We are using an opening curly brace [" { "] to tell Sofya which things we want to start repeating).

- **Line 6:** We are telling the computer to ask the person who is using it, "**What mark did you score in the examination?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Mark** (This line will be repeated if **Count** is not equal to **8**).
- **Line 7:** We are telling the computer to keep on **adding** the marks that someone inputs and we want to store the answer in a variable called **TotalMarks** (This line will be repeated if **Count** is not equal to **8**).
- **Line 8:** We are telling the computer to keep on **adding** the **Count** by **1** and we want to store the answer in a variable called **Count** (This line will be repeated if **Count** is not equal to **8**).
- **Line 9:** We are telling the computer where we want to stop repeating things (That is why we have used a closing curly brace [" } "]).
- **Line 10:** We are telling the computer to check if Variable **Count** is equal to **8**.
- **Line 11:** We are telling the computer that we are making a **variable** called **AverageMark** and that **AverageMark** is **TotalMarks / 8**.
- **Line 12:** We are telling the computer to write what is stored in the variable called **AverageMark** and "**Your average mark is**".
- **Line 13:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

4. Save this Sofya program and then run it.



5. The computer will ask you, "**What mark did you score in the examination?**". Type "50" in the input box and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
50
>>> Program's Output <<<
>>> THE PROGRAM HAS STARTED...
WHAT MARK DID YOU SCORE IN THE EXAMINATION? {Please answer this question in the input box}
```

6. The computer will keep on asking you, "**What mark did you score in the examination?**", seven more times. Type "49", "67", "89", "92", "41", "73" and "65" respectively, in the input box, and press "**Enter**" on the keyboard after you have told the computer a mark scored by Pacifica.
7. If you scroll down the program's output, you will see that the computer will write what is stored in the variable called **AverageMark** and "**Your average mark is**".

```
>>> Program's Output <<<
> What you typed in the input box is: 41
WHAT MARK DID YOU SCORE IN THE EXAMINATION? {Please answer this question in the input box}
> What you typed in the input box is: 73
WHAT MARK DID YOU SCORE IN THE EXAMINATION? {Please answer this question in the input box}
> What you typed in the input box is: 65
YOUR AVERAGE MARK IS 65.75
>>> THE PROGRAM HAS ENDED
```

Rules about Do This...Until Loops

1. The two things that you are comparing in the condition of the loop **can** be:

- A **variable first** then another **variable**, for example,

Until Variable [x] = Variable [b]

- A **variable first** then a **number**, for example,

Until Variable [x] = 24

- A **variable first** then an **expression**, for example,

Until Variable [x] < 22/7

- A **variable first** then a **string**, for example,

Until Variable[x] = "Sofya"

2. If a condition is **true**, the computer will not repeat what you told it to repeat in the curly braces ("{}").
3. You **can** make Do This...Until Loops inside of other Do This...Until Loops.
4. You **can** make **If...Then statements**, **If...Then...Else statements** or **If...Otherwise-If...Else Statements** inside of Do This...Until Loops.

Breaking Loops

Sometimes, you might want to **get out** of a loop (end a loop) before the looping condition in the Do This...Until loop becomes **true**. This is called **breaking a loop**.

Example

In a computer game, a player is required to guess a winning number. If the number guessed is 10, the player is declared a winner and the game ends. If the number guessed is not 10, the player is notified to try again. When the number of trials exceeds 4, the game ends and the player loses. Write a Sofya program to represent the algorithm of this game.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Variable Chance is 0
 - **Line 3:** Do this
 - **Line 4:** {
 - **Line 5:** Ask The Computer User "What is your guess?" and store the answer in Variable[Guess]
 - **Line 6:** If Variable[Guess] = 10 Then
 - **Line 7:** Write "Your answer is correct!" on the screen
 - **Line 8:** Get out of the loop
 - **Line 9:** Otherwise If Variable[Guess] is anything else Then
 - **Line 10:** Write "You are wrong!" on the screen

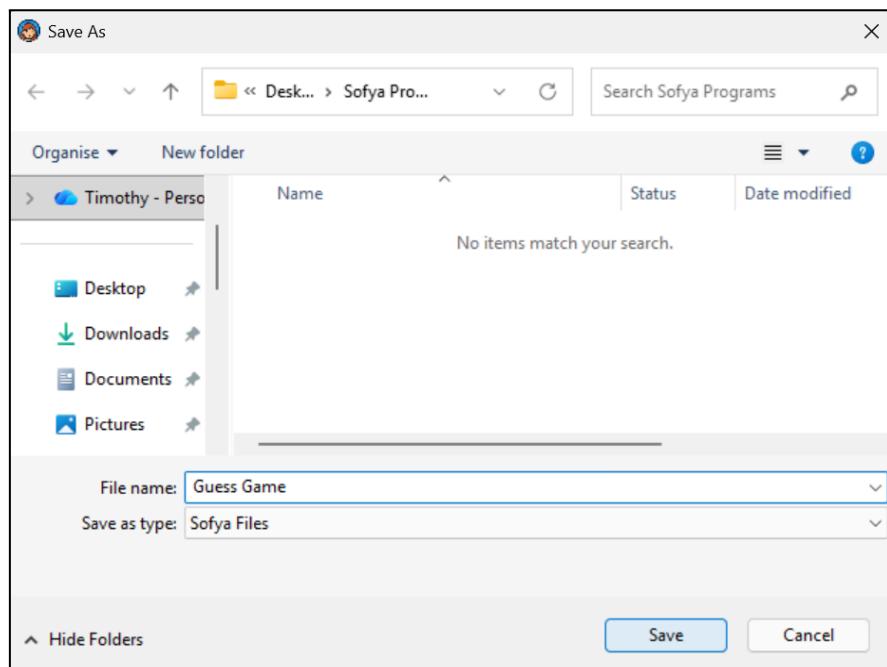
- **Line 11:** Increase Variable[Chance] by 1
- **Line 12:** End this If Statement
- **Line 13:** }
- **Line 14:** Until Variable[Chance] = 4
- **Line 15:** Stop this program

2. Here is an explanation for the program:

- **Line 1:** We are telling the computer that we want to start programming (The program will still work even if you do not say "Start this program").
- **Line 2:** We are telling the computer that we are making a **variable** called **Chance** and that Chance is **0**.
- **Line 3 & 4:** We are telling the computer that we are making a Do This...Until loop (We are using an opening curly brace ["{"] to tell Sofya which things we want to start repeating).
- **Line 5:** We are telling the computer to ask the person who is using it, "**What is your guess?**", and after the person answers this question, we want the computer to store the person's answer in a **variable** called **Guess**.
- **Line 6:** We are telling the computer to check if the **Guess** is equal to **10**.
- **Line 7:** We are telling the computer to write "**Your answer is correct!**" on the screen (This will only happen if the **Guess** is equal to **10**).
- **Line 8:** We are telling the computer to break the loop (This will only happen if the **Guess** is equal to **10**).
- **Line 9:** We are telling the computer to check if the **Guess** is anything else (which is not the correct guess).
- **Line 10:** We are telling the computer to write "**You are wrong!**" on the screen (This will only happen if the **Guess** is not equal to **10**).
- **Line 11:** This means that the computer will keep on adding the **Chance** by **one** every time that someone inputs a wrong guess.
- **Line 12:** We are telling the computer that we want to stop making the If...Then...Else statement (That is why we put "**End this If Statement**").

- **Line 13:** We are telling the computer where we want to stop repeating things (That is why we have used a closing curly brace [" } "]).
- **Line 14:** We are telling the computer to check if Variable **Chance** is equal to Variable **4**.
- **Line 15:** We are telling the computer that we want to stop programming (The program will still work even if you do not say "Stop this program").

3. Save this Sofya program and then run it.



4. The computer will ask you, "**What is your guess?**". You can type "7" and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
7
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR GUESS? {Please answer this question in the input box}
```

5. The computer will write "**You are wrong!**" on the screen then the computer will ask you, "**What is your guess?**". You can type "10" and press "**Enter**" on the keyboard.

```
>>> Input Box <<<
10
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

WHAT IS YOUR GUESS? {Please answer this question in the input box}
> What you typed in the input box is: 7

YOU ARE WRONG!
WHAT IS YOUR GUESS? {Please answer this question in the input box}
```

6. If you scroll down the program's output, you will see that the computer will write "**Your answer is correct!**" on the screen then the program will end.

```
>>> Program's Output <<<

WHAT IS YOUR GUESS? {Please answer this question in the input box}
> What you typed in the input box is: 7

YOU ARE WRONG!
WHAT IS YOUR GUESS? {Please answer this question in the input box}
> What you typed in the input box is: 10

YOUR ANSWER IS CORRECT!

>>> THE PROGRAM HAS ENDED
```

Infinite Loops

Sometimes, you might want the computer to keep on repeating some things as many times as possible until someone **breaks** the **Do This...Until loop**. This is called an **infinite loop**. The general way of writing an infinite loop is:

```
Do this
{
    << The loop body >>
}
Until Infinity
```

Example

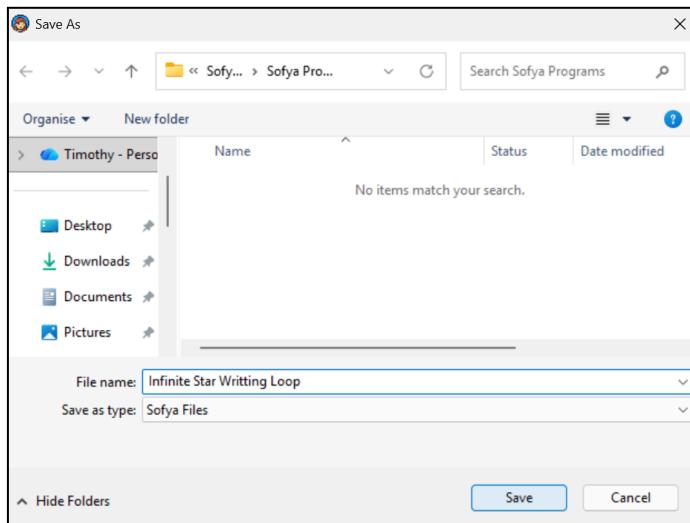
Use Sofya to make a program that asks the computer user whether they want the computer to write **three stars** (***) on the screen. As long as the computer user says

"**Yes**", the computer should continue writing three stars on the screen, but if the computer user says "**No**", the computer should stop writing the three stars on the screen and the program should end.

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Do this
 - **Line 3:** {
 - **Line 4:** Ask The Computer User "Do you want the computer to write 3 stars on the screen?" and store the answer in Variable[Response]
 - **Line 5:** If Variable[Response] = "Yes" Then
 - **Line 6:** Write "***" on the screen
 - **Line 7:** Otherwise If Variable[Response] is anything else Then
 - **Line 8:** Get out of the loop
 - **Line 9:** End this If Statement
 - **Line 10:** }
 - **Line 11:** Until Infinity
 - **Line 12:** Stop this program

2. Save this Sofya program and then run it.



3. The computer will ask you, “**Do you want the computer to write 3 stars on the screen?**”. Type “Yes” in the input box and press **Enter** on the keyboard.

```
>>> Input Box <<<
yes
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
```

4. The computer will write **three stars** (***) on the screen and the computer will again ask you, “**Do you want the computer to write 3 stars on the screen?**”

```
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
> What you typed in the input box is: yes

***

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
```

5. You can repeat **step 3** as many times as you want and you will see that the computer will keep on doing what it did in **step 4**.
6. When the computer asks you, “**Do you want the computer to write 3 stars on the screen?**” and you do not want it to write the 3 stars on the screen, type “**No**” in the input box and press **Enter** on the keyboard.

```
>>> Input Box <<<
no
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
> What you typed in the input box is: yes

***

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
```

7. The computer will not write the 3 stars on the screen and the program will end.

```
>>> Program's Output <<<
>>> THE PROGRAM HAS STARTED...

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
> What you typed in the input box is: yes

***

DO YOU WANT THE COMPUTER TO WRITE 3 STARS ON THE SCREEN? (Please answer this question in the input box)
> What you typed in the input box is: no

>>> THE PROGRAM HAS ENDED
```

Nested Do This...Until Loops

Sometimes, you might want to put a Do This...Until loop **inside** of another Do This...Until loop. When you do this, we say that you are making a **nested Do This...Until loop**.

Example

A bank is organizing a safari rally and racers who are interested are requested to come for a test to see if they would qualify.

In the test:

- Each racer is given a **special number** called a **rally number** that will be used for identification of the racers.
- The racers have to go around a race track for **3 laps**.
- After every lap, the **time taken** to complete a lap is recorded. The time taken to complete a lap is called **lap time**.
- **Lap time** is measured in **seconds**.
- After a racer has finished the 3 laps, the average lap time is calculated. The formula for calculating average lap time is, $L_t(\text{avg.}) = \frac{L_t(\text{total})}{3}$, where $L_t(\text{avg.})$ is the *average lap time* and $L_t(\text{total})$ is the *total lap time for the 3 laps*.

For a racer to qualify for the safari rally, their average lap time should be less than **180 seconds**. If a racer has an average lap time of **180 seconds and above**, then they will **not qualify** for the safari rally. If x racers came for the qualification test of the safari rally, write a Sofya program that can be used to determine whether a racer qualifies for the safari rally or not (**Hint:** Since we do not know how many racers came for the qualification test, we should ask the computer user to tell us how many racers came for the qualification test).

Solution

1. Go to the Sofya Version 1.0 IDLE, and type the following program in the program editor:
 - **Line 1:** Start this program
 - **Line 2:** Variable Racers is 0

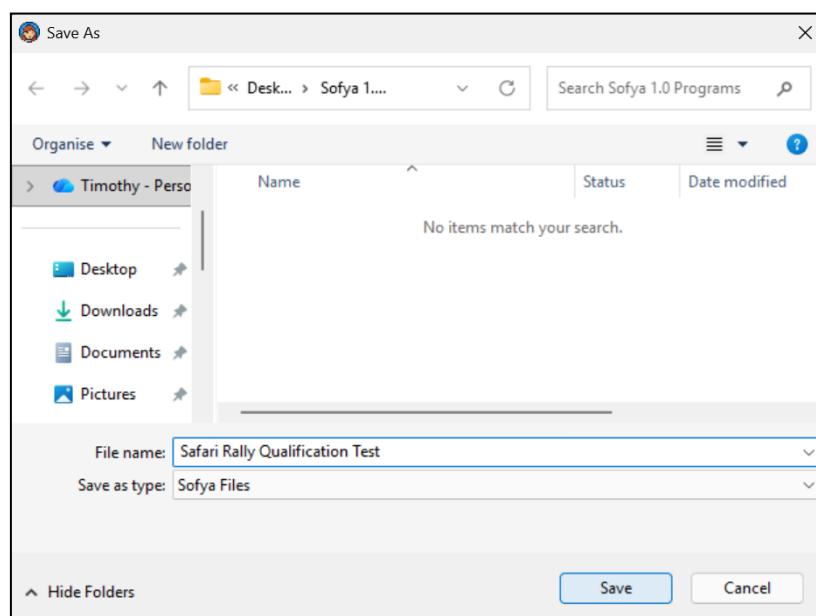
- **Line 3:** Variable Laps is 0
- **Line 4:** Variable TotalLapttime is 0
- **Line 5:** Ask the computer user "How many racers came for the safari rally qualification test?" and store the answer in Variable[TotalRacers]
- **Line 6:** Do this
- **Line 7:** {
- **Line 8:** Ask the computer user "What is the Rally Number of the racer?" and store the answer in Variable[RallyNumber]
- **Line 9:** Make this nested do this until loop
- **Line 10:** {
- **Line 11:** Do this
- **Line 12:** {
- **Line 13:** Ask the computer user "What is the racer's lap time?" and store the answer in Variable[Lapttime]
- **Line 14:** Increase Variable[TotalLapttime] by Variable[Lapttime]
- **Line 15:** Increase Variable[Laps] by 1
- **Line 16:** }
- **Line 17:** Until Variable[Laps] = 3
- **Line 18:** }
- **Line 19:** Variable AverageLapttime is Variable[TotalLapttime] / 3
- **Line 20:** If Variable[AverageLapttime] < 180 Then
- **Line 21:** Write Variable[RallyNumber] and also write "qualified for the Safari Rally with an average lap time of" and also write Variable[AverageLapttime] and also write "seconds" on the screen
- **Line 22:** Otherwise if Variable[AverageLapttime] is anything else Then
- **Line 23:** Write Variable[RallyNumber] and also write "has not qualified for the Safari Rally" on the screen
- **Line 24:** End this If Statement
- **Line 25:** Variable Laps is 0

- **Line 26:** Variable TotalLapttime is 0
- **Line 27:** Increase Variable [Racers] by 1
- **Line 28:** }
- **Line 29:** Until Variable[Racers] = Variable[TotalRacers]
- **Line 30:** Stop this program

2. Here is an explanation of the new concepts:

- **Line 9 and Line 10:** We are telling the computer that we are making a nested Do This...Until loop (We are using an opening curly brace [" { "] to tell the computer where the nested Do This...Until loop starts).
- **Line 18:** We are using a closing curly brace [" } "] to tell the computer where the nested Do This...Until loop ends.

3. Save this Sofya program and then run it.



4. The computer will ask you, “**How many racers came for the safari rally qualification test?**”. You can type “2” in the input box then press **Enter** on the keyboard.

```
>>> Input Box <<<
2
>>> Program's Output <<<
>>> THE PROGRAM HAS STARTED...
HOW MANY RACERS CAME FOR THE SAFARI RALLY QUALIFICATION TEST? (Please answer this question in the input box)
```

5. The computer will ask you, “**What is the rally number of the racer?**”. You can type “**SRN 001**” (Safari Rally Number 001) in the input box then press **Enter** on the keyboard.

```
>>> Input Box <<<
SRN 001
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

HOW MANY RACERS CAME FOR THE SAFARI RALLY QUALIFICATION TEST? {Please answer this question in the input box}
> What you typed in the input box is: 2

WHAT IS THE RALLY NUMBER OF THE RACER? {Please answer this question in the input box}
```

6. The computer will ask you, “**What is the racer’s lap time?**”. You can type “**178.728**” in the input box then press **Enter** on the keyboard.

```
>>> Input Box <<<
178.728
>>> Program's Output <<<

>>> THE PROGRAM HAS STARTED...

HOW MANY RACERS CAME FOR THE SAFARI RALLY QUALIFICATION TEST? {Please answer this question in the input box}
> What you typed in the input box is: 2

WHAT IS THE RALLY NUMBER OF THE RACER? {Please answer this question in the input box}
> What you typed in the input box is: SRN 001

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
```

7. The computer will again ask you, “**What is the racer’s lap time?**”. You can type “**181.997**” in the input box then press **Enter** on the keyboard.

```
>>> Input Box <<<
181.997
>>> Program's Output <<<
HOW MANY RACERS CAME FOR THE SAFARI RALLY QUALIFICATION TEST? {Please answer this question in the input box}
> What you typed in the input box is: 2

WHAT IS THE RALLY NUMBER OF THE RACER? {Please answer this question in the input box}
> What you typed in the input box is: SRN 001

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 178.728

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
```

8. The computer will again ask you, “**What is the racer’s lap time?**”. You can type “**173.119**” in the input box then press **Enter** on the keyboard.

```
>>> Input Box <<<
173.119
>>> Program's Output <<<
WHAT IS THE RALLY NUMBER OF THE RACER? {Please answer this question in the input box}
> What you typed in the input box is: SRN 001

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 178.728

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 181.997

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
```

9. The computer will tell you whether the racer has qualified for the safari rally or not, then the computer will ask you “**What is the rally number of the racer?**”.

```
>>> Program's Output <<<
WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 181.997

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 173.119

SRN 001 QUALIFIED FOR THE SAFARI RALLY WITH AN AVERAGE LAP TIME OF 177.948 SECONDS

WHAT IS THE RALLY NUMBER OF THE RACER? {Please answer this question in the input box}
```

10. You can **repeat** steps 5 - 8 but you can give the computer the following details about the second racer:

- **Rally Number:** SRN 002.
- **Lap time for the 1st lap:** 189.913 seconds.
- **Lap time for the 2nd lap:** 182.764 seconds.
- **Lap time for the 3rd lap:** 179.991 seconds.

11. The computer will tell you whether the racer has qualified for the safari rally or not, then the program will end.

```
>>> Program's Output <<<
> What you typed in the input box is: 189.913

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 182.764

WHAT IS THE RACER'S LAP TIME? {Please answer this question in the input box}
> What you typed in the input box is: 179.991

SRN 002 HAS NOT QUALIFIED FOR THE SAFARI RALLY

>>> THE PROGRAM HAS ENDED
```

Program Errors in Sofya

The table below shows the **names**, **causes**, **solutions** and **examples** of program errors that you can find in Sofya Version 1.0:

Table of Program Errors

Name of the error	Cause of the error	Solution to the error	Example of the error
Variable Error (or Unknown Variable Error)	You have told Sofya to do something with a variable and you have not yet made that variable	Make the variable first and then you can tell Sofya what to do with the variable	You might want Sofya to write a variable called ' Volume ' on the screen but you did not tell Sofya what to store in the variable called 'Volume'
Variable Storage Error	You have told Sofya to store things in a variable in a way that does not make sense	Only store things in a variable that are meant to be stored together	You might say Variable Area is Constant[Pi] x "Seven" exp 2
Empty Variable Error	You have made a variable but you have not stored anything in that variable	After you make a variable, you should store something in that variable	You might say Variable Volume is
New File Error	You have told the computer to make a new Sofya file but you are already using a new Sofya file	Just ignore the error since you are already using a new file	You have told the computer to make a new Sofya file but you are already using a new Sofya file
Syntax Error	In the Sofya program, you have used something that does not follow the rules of Sofya	Make sure that your Sofya program follows the rules of Sofya	You might tell the computer to write ' hello world ' by saying, Write hello world
Division by Zero Error	There is a place in the Sofya program where a number is divided by zero	Make sure that your Sofya program does not have a place where a number is divided by zero	You might tell the computer to do, Variable Answer is 3/0

Note: Here are some other program errors that you can find in Sofya Version 1.0:

Loop Error

A loop error happens when you do **not** make all the **Do this...Until loops** properly (The computer usually shows you the loop error on the **program's output** of the Sofya Version 1.0 IDLE). For example, let us look at this Sofya program:

```
Start this program
    Variable Counter_1 is 0
    Variable Counter_2 is 0
    Do this
    {
        Write "This is the outer loop" on the screen
        Make this nested do this until loop
        {
            {
                Write "Hello World" on the screen
                Increase Variable[Counter_2] by 1
            }
            Until Variable[Counter_2] = 3
        }
        Increase Variable[Counter_1] by 1
        Variable Counter_2 is 0
    }
    Until Variable[Counter_1] = 10
Stop this program
```

This program will cause a loop error because *the part that is highlighted is a Do this...Until loop but we never said "Do this" before we started making this loop*. So, the computer will **write "Hello World"** 3 times (because of the line "**Until Variable[Counter_2] = 3**") but when it reaches the line "**Until Variable[Counter_1] = 10**" the computer will show you a loop error message on the program's output in the Sofya Version 1.0 IDLE.

IDLE Crash

An **Integrated Development and Learning Environment (IDLE) crash** happens when the Sofya Version 1.0 IDLE hangs (or lags). An IDLE crash happens when you

forget to **break** an **infinite Do this...Until loop**. It can also happen when there is a **loop error**. When an IDLE crash happens, you can just close (or exit) the Sofya Version 1.0 IDLE and then you can open it again and then break the infinite Do this...Until loop or you can correct the place where there is a loop error.

References used in developing Sofya

Here is a list of all the references that were used to do research to develop Sofya Version 1.0:

1. Matematickcom Official. (2022, January 15). *The first female math Ph.D. who broke all barriers - Sofya Kovalevskaya* [Video]. YouTube.
https://youtu.be/x2Ot_YuQhCw?si=W5mFAzyOOBHLyeWG
2. howCode. (2014, June 29). *Make your own programming language - Introduction* [Video]. YouTube. <https://youtu.be/pWAXiKdJF0c?si=AgN8uWbOOn9w7sDI>
3. Programming Hero. (2021, March 22). *Create your own IDE || Build a Python compiler || Best IDE for Python || Python project* [Video]. YouTube.
<https://www.youtube.com/watch?v=f1u3me4GYmw&list=PPSV&t=5s>
4. CodersLegacy. (2023, January 4). *Change Tkinter window icon - (Two different methods!)* [Video]. YouTube.
<https://www.youtube.com/watch?v=s1-leIK9LWc&list=PPSV>
5. Hamedani, M. [Programming with Mosh]. (2019, February 18). *Python full course for beginners* [Video]. YouTube.
https://youtu.be/_uQrJ0TkZlc?si=7UbZOr5iODBcZ6n3
6. Stratvert, K. [Kevin Stratvert]. (2021, March 25). *Python for beginners tutorial* [Video]. YouTube. <https://youtu.be/b093aqAZiPU?si=uSiIYdePvQl34Xmx>
7. Wikipedia. (2024, September 26). *List of physical constants*. [List of physical constants - Wikipedia](#)
8. Wikipedia. (2025, December 23). *List of mathematical constants*. [List of mathematical constants - Wikipedia](#)
9. Mburu, S., & Chemwa, G. (2012). *Computer Studies Form 3* (2012 Edition). Longhorn Kenya. [Longhorn Computer Studies F3](#)

10. Kams KAI. (2024, November 17). *How to install Python on Windows 11, installing Python on Windows 11* [Video]. YouTube.
<https://youtu.be/noGdD3w6zdE?si=6ecZmVKTpel17NBR>
11. Kams KAI. (2024, November 20). *How to install pip in Windows 11 - Install pip in Python on Windows* [Video]. YouTube.
<https://youtu.be/qGD1s1CFv1I?si=3IbKLqvrM3cLs39b>
12. Elder, J. [[Codemy.com](#)]. (2020, February 18). *Standalone Python EXE Executable - Python Tkinter GUI Tutorial #40* [Video]. YouTube.
<https://youtu.be/QWqxRchawZY?si=-8EC1XNuvdvJ6s5v>
13. Code First with Hala. (2022, September 16). *Convert Tkinter Python App to Executable (.exe) File [pyinstaller]* [Video]. YouTube.
<https://youtu.be/IvdECetoM?si=5HDN9uDtvAfIhIaf>

Conclusion

Thank you for reading this manual! I hope you found it useful. If you have any questions, please send an email to oyweraa@gmail.com or you can reach out to me on GitHub using this link <https://github.com/oyweraa-hue/Sofya-Version-1.0> and I will be happy to help. Have fun using Sofya!