**Python:As a scripting language**

Subject:- Unix Operating System System Lab Class :- TYIT

**Name: - SHRENIK JADHAV**

**OM GHARGE**

**PRN: -2020BTEIT00011,2020BTEIT00041**

**Assignment No 10a**

**Title-** Write a program to display the following pyramid. The number of lines in the pyramid should not be hard-coded. It should be obtained from the user. The pyramid should appear as close to the center of the screen as possible.

**Objective:**

1. To learn about fundamentals of IPC through C socket programming.
2. Learn and understand the OS intraction with socket programming.
3. Use of system call and IPC mechanism to write effective application programs.
4. To know the port numbersing and process relation
5. To knows the iterative and concurrent server concept

**Theory:**

Subprocess Management:

The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions:

os.system os.spawn\* os.popen\* popen2.\* commands.\*

Using the subprocess module:

The recommended way to launch subprocesses is to use the following convenience functions. For more advanced use cases when these do not meet your needs, use the underlying Popen interface. subprocess.call(args, \*, stdin=None, stdout=None, stderr=None, shell=False)

Run the command described by args. Wait for command to complete, then return the returncode attribute.

The arguments shown above are merely the most common ones, described below in Frequently Used Arguments (hence the slightly odd notation in the abbreviated signature). The full function signature is the

same as that of the Popen constructor - this functions passes all supplied arguments directly through to that interface.

**Program-**

#include <stdio.h>

void displayPyramid(int numLines) {

int spaces, stars;

for (int i = 1; i <= numLines; i++) {

spaces = numLines - i;

// Print spaces

for (int j = 1; j <= spaces; j++) {

printf(" ");

}

stars = 2 \* i - 1;

// Print stars

for (int k = 1; k <= stars; k++) {

printf("\*");

}

printf("\n");

}

}

int main() {

int numLines;

printf("Enter the number of lines in the pyramid: ");

scanf("%d", &numLines);

printf("\n");

displayPyramid(numLines);

printf("\n");

return 0;

}

run--

Open a text editor and save the code into a file, for example, pyramid.c.

* Open a terminal and navigate to the directory where you saved the file.
* Compile the program using the gcc compiler. Run the following command in the terminal:
* gcc pyramid.c -o pyramid
* After successful compilation, you can execute the program by running:

bash

* ./pyramid
* The program will prompt you to enter the number of lines in the pyramid. Provide the desired value and press Enter.
* The program will display the pyramid, centered on the screen as much as possible.

**Output-**

Enter number of rows:6

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

**Conclusion-**

1. .Basics of python like the concept of loops learnt
2. .Conditional statements learn

**Reference-**

https://docs.python.org/3/