PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Ft. Road, BSK III Stage, Bengaluru – 560 085

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Title: Problem Solving with C Laboratory		
Course code: UE19CS152		
Semester : II sem	**Section: Q**	**Team Id: 5**
SRN: PES1UG19CS056	**Name: Amritesh**	
SRN: PES1UG19EC243	**Name: Rounak B**	
SRN: PES1UG19CS512	**Name: Sudhanva Rajesh**	
SRN: PES1UG19CS562	**Name: Vemula Vaibhav**	

PROJECT REPORT

Problem Statement:

The Project deals with conversion of code written in programming language C to programming language Python.

Description:

GOALS & OBJECTIVES:

- With this project we want to create a program which can easily accept basic C codes and give an equivalent code in Python.
- This project tries to successfully provide a working Python code with all proper syntax and indentation.

PROBLEMS TO BE ADDRESSED:

- Basic C code has different header files that are not there in Python and thus will give an error so we tried to remove them.
- Moreover, python doesn't need to work on main() function so we had to put all the main() function code separately which was successfully done.
- Python has a lot of same keywords like if, else but at the same time it has different keywords for some of the functions like printf() becomes print() in Python thus it had to be taken care of.
- In C the declaration of variable is totally different and working of printf and scanf is also different and also the working of loops such as for loop is different.
- One of the important thing in Python is indentation and if we do not provide it our code can go wrong in many ways and thus it was also taken care of.

APPROACHES:

- We have tried to put forward this project in just one file.
- Functions have been used to give a proper output.
- The functions are being called whenever their required conditions are being fulfilled.
- Moreover, Input is being taken in the terminal itself and as we get an input for one line it calls the required function.
- The final output is being stored in a string and later displayed in the terminal.

C-concepts used:

Functions:

Functions have been used to do any work on the input taken thus function plays a very important role.
SYNTAX:

```
void convert(char code[],char code2[],char check[]){}
```

Arrays & Strings:

As we are aware that there doesn't exist any datatype as String in C. So, storing information was a problem. Arrays were used to store information. Moreover, because we are taking input in terminal itself instead of using file handling so everything was required to be saved in an array as a string so basic array iterations were also used.
Concepts of appending any value to an array were also used because the main array was always appending new values.

getchar() & putchar():

getchar() and putchar() is used to take input for just one character and thus that idea has been used.

scanf() & printf() Concepts:

scanf() and printf() uses format specifiers which is not used in Python. So, we need to remove those and put the required variable or literal. Moreover, a particular format specifier is used for a particular datatype and thus it had to be dealt with in the project.
Different format specifiers like: `%f, %d, %ld`

Loops & nested Loops:

Loops and nested Loops have been used extensively in the project because when manipulating strings they were required and also to check a certain conditions.

Nested Functions:

Nested functions have been used in the program to make the program more sorted and working and easy to use.

String Functions:

String functions like strlen() have been used to check the length of a string for a particular purpose.
e.g. `strlen(code2); //where code2 is a string.`

Other Basic C Concepts:

Other basic C Concepts like printf and scanf and if/else has been used. Nested if else has also been used in the program. Ideas of local and global variable has also been used.

Learning Outcome:

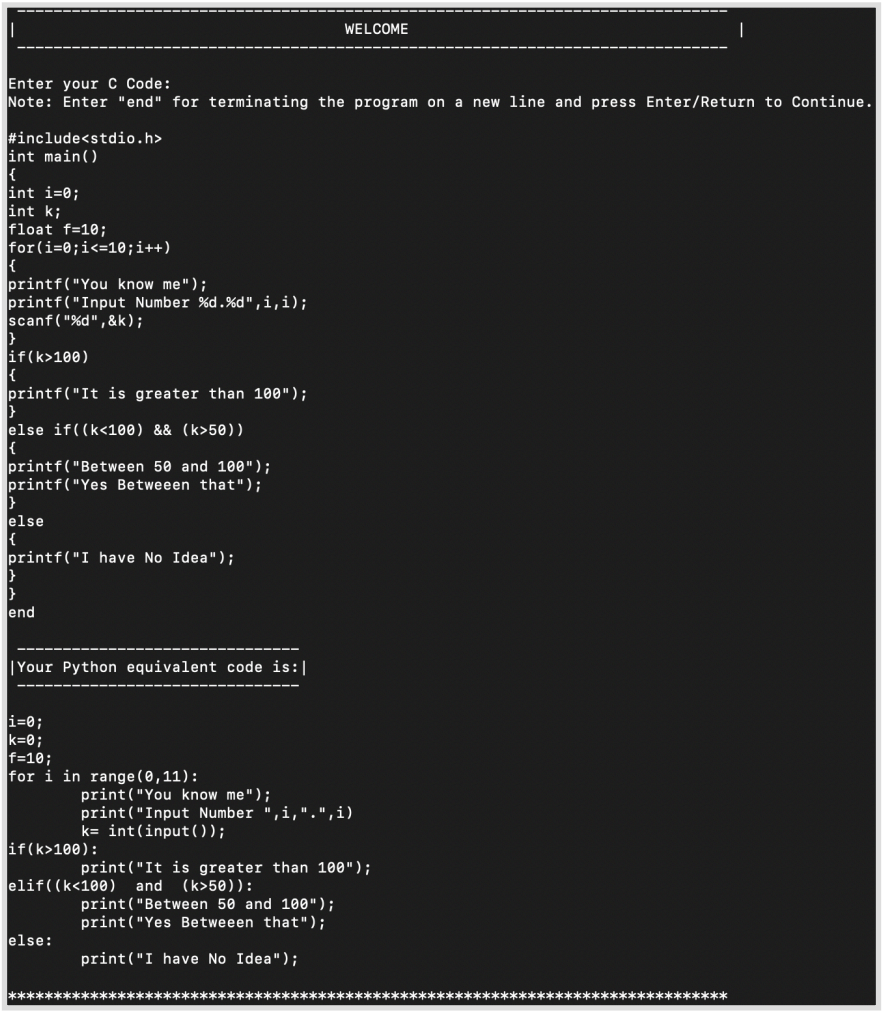
Self-Learning Component:

- How to deal with string and manipulate them according to our needs.
- Calling of different conditional statements in the project and then working according to them.
- Local and global variable and how to call them in a function.
- Different types of errors and how to deal with them(like bus error and segmentation fault, etc).
- Different level of complexities of both the programming language i.e. C and Python
- Different keywords and how they are different in Python and C.

Observations & Outcomes:

- C works differently on different on systems.
- Working with strings in C is difficult because there is no datatype as string.
- Sometimes, the output differs on different systems.
- Moreover, how to deal with strings and manipulate them and while trying for file handling how input is taken by C.

Output Screenshots:



```
 ------------------------------------------------------------------------------
|                                  WELCOME                                     |
 ------------------------------------------------------------------------------

Enter your C Code:
Note: Enter "end" for terminating the program on a new line and press Enter/Return to Continue.

#include<stdio.h>
int main()
{
int i=0;
int k;
float f=10;
for(i=0;i<=10;i++)
{
printf("You know me");
printf("Input Number %d.%d",i,i);
scanf("%d",&k);
}
if(k>100)
{
printf("It is greater than 100");
}
else if((k<100) && (k>50))
{
printf("Between 50 and 100");
printf("Yes Betweeen that");
}
else
{
printf("I have No Idea");
}
}
end

 ------------------------------
|Your Python equivalent code is:|
 ------------------------------

i=0;
k=0;
f=10;
for i in range(0,11):
        print("You know me");
        print("Input Number ",i,".",i)
        k= int(input());
if(k>100):
        print("It is greater than 100");
elif((k<100)  and  (k>50)):
        print("Between 50 and 100");
        print("Yes Betweeen that");
else:
        print("I have No Idea");

********************************************************************************
```

Name and Signature of the Faculty: