PYTHON PROGRAMMING (INT213)

MINI PROJECT ON GUI INTERFACE TO CONVERT NUMBER FROM ONE NUMBER SYSTEM TO OTHER NUMBER SYSTEM

Submitted By:

Ayush Saini (12114732)

Sujiith S (12114735)

Pratham Thakur (12114778)

K21MR (BTECH CSE)

Submitted To:

Dr. Ramandeep Sandhu

School of Computer Science and Engineering Lovely Professional University

INTRODUCTION:

The arithmetic value which is used for representing the quantity and used in making calculations are defined as NUMBERS. The writing system for denoting numbers using digits or symbols in a logical manner is defined as a Number system. Number System is a system that defines numbers in different ways to represent numbers in different forms.

Types of Number System

The number system in python is represented using the following four systems:

- Binary Number System (base or radix =2)
- Octal Number System (base or radix = 8)
- Decimal Number System (base or radix = 10)
- Hexadecimal Number System (base or radix = 16)

Binary Number System

A number system with base or radix 2 is known as a binary number system. Only 0 and 1 are used to represent numbers in this system.

Octal Number System

Octal Number System is one in which the base value is 8. It uses 8 digits i.e. 0-7 for the creation of Octal Numbers. It is also a positional system i.e weight is assigned to each position.

Decimal Number System

A number system with a base value of 10 is termed a Decimal number system. and it is represented using digits between (0 to 9). Here, the place value is termed from right to left as first place value called units, second to the left as Tens, so on Hundreds, Thousands, etc.

Hexadecimal Number System

A number system with a base of 16 is called a hexadecimal number system. It is represented using numbers between 0 to 9 and the alphabets between A to F. As both numeric digits and alphabets are used in this system, it is also called an alphanumeric system.

OBJECTIVE:

The objective of this project is to Develop a GUI interface to convert number form one number system (binary, octal, decimal and hexadecimal) to other number system (binary, octal, decimal and hexadecimal) with explanation using Python.

MODULE WISE DESCRIPTION:

➢ **Module 1 –** Convert any number to Decimal number

➢ **int()** is the function used to convert any number into Decimal number

Case 1 – (Binary to Decimal)
0b is the prefix used to denote Binary number
Pseudocode –
>>>int(0b1010)
>>>10

Case 2 – (Octal to Decimal)
0o is the prefix used to denote Octal number
Pseudocode –
>>>int(0o1010)
>>>520

Case 3 – (Hexadecimal to Decimal)
0x is the prefix used to denote Hexadecimal number
Pseudocode -
>>>int(0x1010)
>>>4112

➢ **Module 2 –** Convert any number to Binary number

➢ **bin()** is the function used to convert any number into Binary number

Case 1 – (Decimal to Binary)
There's no specific prefix for Decimal number
Pseudocode –
>>>bin(1010)
>>>'0b1111110010'

Case 2 – (Octal to Binary)
Pseudocode –
>>>bin(0o1010)
>>>'0b1000001000'

Case 3 – (Hexadecimal to Binary)
Pseudocode –
>>>bin(0x1010)
>>>'0b1000000010000'

➢ **Module 3 –** Convert any number to Octal number

➢ **oct()** is the function used to convert any number into Octal number

Case 1 – (Decimal to Octal)
Pseudocode –
>>>oct(1010)
>>>’0o1762’

Case 2 – (Binary to Octal)
Pseudocode –
>>>oct(0b1010)
>>>'0o12'

Case 3 – (Hexadecimal to Octal)
Pseudocode –
>>>oct(0x1010)
>>>'0o10020'

➢ **Module 4 –** Convert any number to Hexadecimal

➢ **hex()** is the function used to convert any number into Hexadecimal number

Case 1 – (Decimal to Hexadecimal)
Pseudocode –
>>>hex(1010)
>>>'0x3f2'

Case 2 – (Binary to Hexadecimal)
Pseudocode –
>>>hex(0b1010)
>>>'0xa'

Case 3 – (Octal to Hexadecimal)
Pseudocode –
>>>hex(0o1010)
>>>'0x208'

Decimal to Other Base System

Steps

- **Step 1 −** Divide the decimal number to be converted by the value of the new base.
- **Step 2 −** Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
- **Step 3 −** Divide the quotient of the previous divide by the new base.
- **Step 4 −** Record the remainder from Step 3 as the next digit (to the left) of the new base number

ROLES AND RESPONSIBILITIES:

Pratham Thakur (12114778) - Responsible for creating user interface.
user interface- a computer program that enables a person to communicate with acomputer through the use of symbols, visual metaphors, and pointing devices.

Ayush Saini (12114732) and Sujiith S (12114735) – responsible for creatingmodules.
Modules - A module is a file containing python definitions and statements. A modulecan define functions, classes, and variables. A module can also include runnable code.Grouping related code into a module makes the code easier to understand and use.

CODE WISE DESCRIPTION:

1) Setting up the Title

main.pyw - X:\New folder\main.pyw (3.10.7)

File Edit Format Run Options Window Help

```
from tkinter import Tk, Label, Frame, X, StringVar, LEFT, Entry, RIGHT, Button
from tkinter import ttk
import convertBase

root = Tk()
root.maxsize(610, 355)
root.minsize(590, 345)
root.geometry(f"{600}x{350}")
root.title("Numerical Base Converter")
```

2) Creating drop boxes

```
bases = [2,8,10,16]
ttk.Style().configure("style1.TCombobox", foreground="blue", background="black")

def createDropBox(textVariable, textToDisplay):
    frame2 = Frame(root)
    frame2.pack(anchor = "center", pady = 10)
    Label(frame2, text = textToDisplay).pack(side = LEFT)
    baseFrom = ttk.Combobox(frame2,  width = 40, height = 20, textvariable = textVariable, values = bases, style = "style1.TCombobox", font = "ms-sans 15")
    baseFrom.pack(side = RIGHT)
```

3) Calling convert base module

```
def calculate():
    global calculation
    converted = convertBase.convert(number.get(), baseFrom.get(), baseTo.get())
    try:calculation.pack_forget()
    except Exception:pass
    calculation = Label(root, text = converted, bg = "dark blue", fg = "white", pady = 57, font = "ms-sans 20")
    calculation.pack(anchor = "center", fill = X)
```

4) To check if the number is correct or not

w folder > convertBase.py

```
correspondenceString = r"""0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz/-"""

def convert(number = '10', baseFrom = 10, baseTo = 10):
    check = checks(number, baseFrom, baseTo)
    if not check.accepted:return check.message
    convertedToBase10 = convertToBase10(number, baseFrom)
    return convertToBase(convertedToBase10, baseTo)
```

5) To convert it into Base10

```
def convertToBase10(number = "10", baseFrom = 10, correspondenceString = correspondenceString):
    baseFrom = int(baseFrom)
    charToNumCorrespondence = {j:i for i, j in enumerate(correspondenceString)}
    convertedBase10 = 0

    for num, char in enumerate(reversed(str(number))):
        convertedBase10 += int(charToNumCorrespondence[char]) * (baseFrom ** num)

    return convertedBase10
```

6) To convert into any base

```python
def convertToBase(number = "10", baseTo = 10, correspondenceString = correspondenceString):
    baseTo = int(baseTo)
    numToCharCorrespondence = {i:j for i, j in enumerate(correspondenceString)}
    convertedNum = ""

    while number != 0:
        remainder = number % baseTo
        corresponding = numToCharCorrespondence[remainder]
        convertedNum += str(corresponding)
        number //= baseTo

    return convertedNum[::-1]
```

7) To check any errors

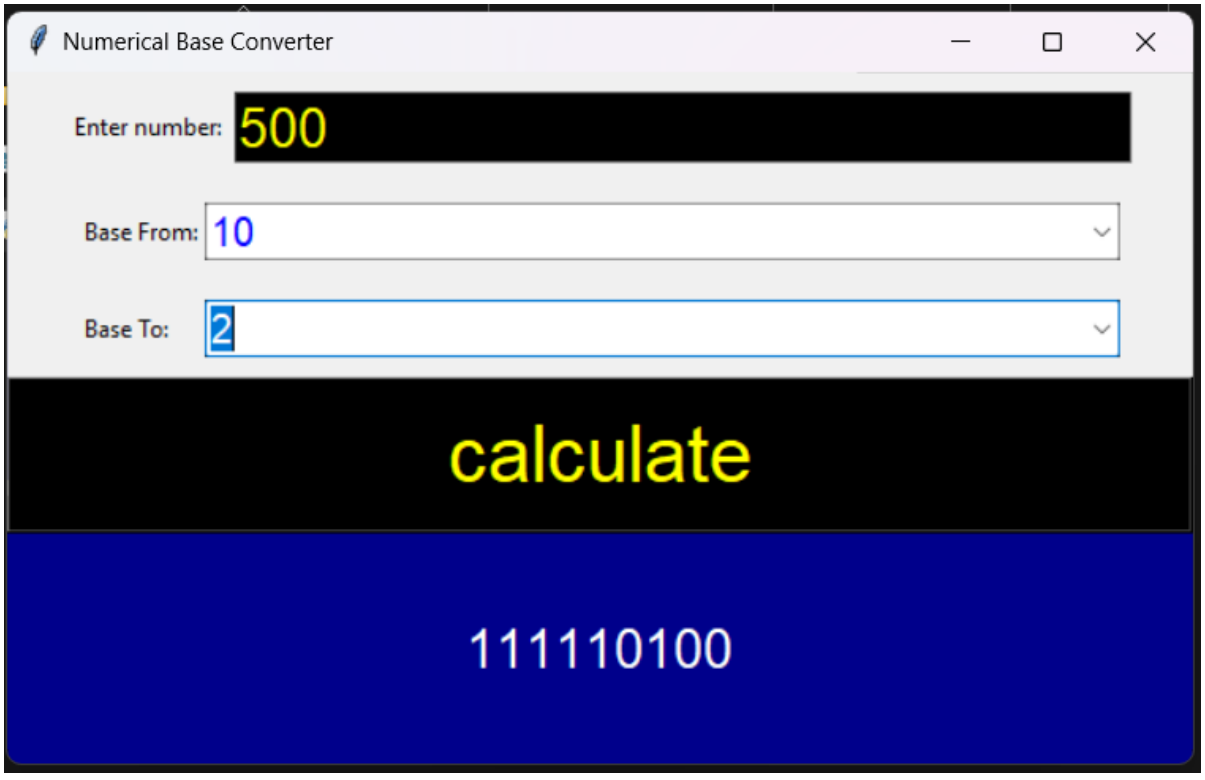
```python
def checks(number, baseFrom, baseTo, correspondenceString = correspondenceString):
    try:
        baseTo = int(baseTo);baseFrom = int(baseFrom)
    except Exception:return checkMessage(False, "invalid base, must be a numerical value")

    if baseTo < 1 or baseFrom < 1:return checkMessage(False, "base must be greator then 0")
    if baseTo > len(correspondenceString):return checkMessage(False, "this base is not supported")
    for i in str(number):
        if i not in correspondenceString:return checkMessage(False, "invalid charater")

    return checkMessage(True, "All parameters accepted")


class checkMessage:
    def __init__(self, accepted, message):
        self.accepted, self.message = accepted, message
```

RESULT SCREENSHOTS:



CONCLUSION:

On implementing this project, we learnt the technology Python-GUI. We also learnt the details about the number system conversion. The skills which we have gained from this project have been immense they have helped us gain confidence in our ability to make new programmes

REFERENCE:

https://www.tutorialspoint.com/computer_logical_organization/binary_codes.htm

https://note.nkmk.me/en/python-bin-oct-hex-int-format/

https://www.programiz.com/python-programming/examples/conversion-binary-octal-hexadecimal

https://www.geeksforgeeks.org/number-system-in-python/