1. **Made a few grammatical and spelling corrections to the previous version.**
2. def GetStringWithoutNumbers(dna1):

line=dna1

line1=[]

for i in range (len(line)-1):

base=line[i]

if base == 'a' or base =='A':

line1 += ['a']

elif base == 't' or base =='T' or base == 'u' or base == 'U':

line1 += ['t']

elif base == 'c' or base == 'C':

line1 += ['c']

elif base == 'g' or base == 'G':

line1 += ['g']

else: #Edit 2023

messagebox.showerror("Error!", "Your coding sequence has invalid characters. \nInput valid characters (A,U,G,T,C) and BioBrick again.")

break

**Included an error box to notify user of any invalid character, which is, any character other than A,U,T,G,C.**

1. def clear2(): #Edit 2023

global txt4

global lbl\_warn

global lbl\_help

lbl\_warn.pack()

lbl\_warn.pack\_forget()

lbl\_help.pack()

lbl\_help.pack\_forget()

txt4.pack()

txt4.pack\_forget()

txt3.place(x = 30, y = 340 , width=700, height=150)

txt3.delete("1.0","end")

lbla.configure(text="")

lbl3.configure(text="")

lbl4.configure(text="")

lbl5.configure(text="")

lbl6.configure(text="")

lbl7.configure(text="")

lbl8.configure(text="")

lbl\_com.configure(text="")

lbl\_com2.configure(text="")

btn2=Button(window, text="Clear output", bg="white", fg="black", command=clear2) #Edit 2023

btn2.place(x = 630, y = 540, width = 80)

btn1=Button(window, text="Clear input", bg="white", fg="black", command=clear1)

btn1.place(x = 540, y = 540, width = 80)

**Included clear output and clear input buttons separately.**

1. def copy(): #Edit 2023

if txt4.get("1.0", "end-1c"):

bbseq = txt4.get("1.0",END)

bbseq = str(bbseq)

window.clipboard\_clear()

window.clipboard\_append(bbseq)

else:

bbseq = txt3.get("1.0",END)

bbseq = str(bbseq)

# print(bbseq)

window.clipboard\_clear()

window.clipboard\_append(bbseq)

btn\_copy = Button(window, textvariable=button\_text, bg="white", fg="black", command=copy)

btn\_copy.place(x = 550, y = 510, width = 150)

**Updated code to copy either BioBricked, reverse BioBricked sequence or suggested sequence if the provided sequence could not be BioBricked.**

1. txt3.tag\_add('tag10','1.0 + 14 chars','1.0 + 22 chars') #Edit2023

txt3.tag\_config('tag10', foreground="purple")

txt3.tag\_add('tag11','end - 22 chars','end - 15 chars') #Edit2023

txt3.tag\_config('tag11', foreground="purple")

lbl8.configure(text='NotI', foreground='purple') #Edit2023

lbl5.configure(text='SpeI', foreground='brown') #Edit2023

**Included NotI in the legend and highlighted the sequences in the BioBricked and reverse BioBricked sequence. Changed color of SpeI to brown for better visibility.**

1. # Copy\_Button for Version 3

button\_text = StringVar()

def update\_button\_text(): #Edit 2023

if txt4.get("1.0", "end-1c"):

button\_text.set("Copy suggested sequence")

else:

button\_text.set("Copy BioBrick sequence")

window.after(10, update\_button\_text)

btn\_copy = Button(window, textvariable=button\_text, bg="white", fg="black", command=copy)

btn\_copy.place(x = 550, y = 510, width = 150)

txt4.bind("<KeyRelease>", lambda e: update\_button\_text())

txt3.bind("<KeyRelease>", lambda e: update\_button\_text())

update\_button\_text()

**Changed text of copy button to "Copy suggested sequence" if there is an invalid sequence.**