LOYOLA ACADEMY

DEGREE & PG COLLEGE

OLD ALWAL, SECUNDERABAD - 500 010,
TELANGANA, INDIA

An Autonomous Institution Affiliated to Osmania University

Re-accredited with ‘A’ Grade (III Cycle) by NAAC A

“College with Potential for Excellence” by UGC

Practical Record

CERTIFICATE

This is to certify that this is a Bonafide record work done in Python Scripting practical during II year IV semester of the academic year 2024-2025

Name:

UID No:

Class:

Signature of Internal **Signature of HoD**

Signature of External **Signature of Principal**

INDEX

18.	Slicing an array		
19.	Searching element in an array		
20.	Updating elements in an array		
21.	Counting elements in array		
22.	Reversing elements in an array		
23.	Extend elements from array		
24.	Program to display Check-box		
25.	Program to display Label		
26.	Program to display Buttons		
27.	Program to display Radio-buttons		
28.	Program to display Dialogue-box		
29.	Program to display Radio-buttons II		
30.	Program to display creation of Home		
31.	Program to display Menu		
32.	Program using Radio buttons with Submit button		
33.	Program using Check box to display any three categories with Submit button		
34.	Program to display message using Dialogue box		
35.	Using arrays to display the student grades		
36.	Using arrays to find whether the student has passed or failed		

37.	Using arrays to display the number of Books, Publisher name and Year Of Publication		
38.	XML program to create the library information		
39.	Program to demonstrate DOM		
40.	Program to save the XML file as HTML file		

1) Programs on regular expressions

```
import re

s='Welcome to cyber security 2nd year'

match =re.search(r'cyber',s)

print('Start index:',match.start())

print('End index:',match.end())
```

OUTPUT:-

```
Start index: 11

End index: 16
```

2) To find all the characters in the string that fall within the range of 'a' to 'm' using find all the methods

```
import re

string="The quick brown fox jumps over the lazy dog"

pattern="[a-m]"

result=re.findall(pattern,string) print(result)
```

OUTPUT:-

```
['h', 'e', 'i', 'c', 'k', 'b', 'f', 'j', 'm', 'e', 'h', 'e', 'l', 'a', 'd', 'g']
```

3) To check if a list of strings are matched or unmatched

```
import re

regex=r'The'

strings=['The quick brown fox', 'The lazy dog','A quick brown fox'] for

string in strings:

if        re.match(regex,string):

print(f'matched:{string}')

else:

print(f'Not matched:{string}')
```

OUTPUT:-

```
matched:The quick brown fox matched:The

lazy dog

Not matched:A quick brown fox
```

4) To check if the string ends with "World!"

```
import re string="Hello

World!"

pattern=r"World!$"

match=re.search(pattern,string) if

match:

print("Match found!") else:

print("Match not found.")
```

OUTPUT:-

```
Match found!
```

5) To search for the pattern "brown.fox" within the string.

```
import re

string="The quick brown fox jumps over the lazy dog."

pattern=r"brown.fox"

match=re.search(pattern,string) if

match:

print("Match found!") else:

print("Match not found.")
```

OUTPUT:-

```
Match not found.
```

6) python programming using numbers in pyhton

```
a=4

b=4.5

c=2j

print       (type(a))

print       (type(b))

print (type(c))
```

Output:

```
<class 'int'>

<class 'float'>

<class 'complex'>
```

7) using arithematic opertaions on int type

```
res=5+3 print

(res) res=5-3

print (res)

res=5*3 print

(res) res=15/4

print (res)

res=15//4 print

(res) res=15%4

print (res)

res=2**3 print

(res) res=abs(-

10) print (res)

res=round(3.14159,2)

print (res)
```

Output:

```
8        3

2        8

15       10

3.75     3.14

3
```

8) Using arithematic opertaions on complex type

```
res=(3+4j)+(1+2j)

print (res)

res=(5+6j)-(2+3j)

print (res)

res=(2+3j)*(1+4j)

print (res)

res=(8+6j)/(2+3j)

print (res)

res=(1+1j)**2 print

(res) res=abs(3+4j)

print (res)

res=(3+4j).conjugate()

print (res) real=(3+4j).real

imag=(3+4j).imag print

(real)

print (imag)
```

Output:

```
(4+6j)                                              2j          4.0

(3+3j)                                              5.0

(-10+11j)                                           (3-4j)

(2.6153846153846154-0.9230769230769231j)            3.0
```

9) Using built in functions

```
a=2

print(float(a)

b=5.6

print(int(b))

c=‘3’

print(type(int(c)) d=’5.6’

print(type(float(d)) e=5

print(complex(e)) f=6.5

print(complex(f))
```

Output:

```
2.0

5

<class ‘int’>

<class ‘float’>
(5+0j)

(6.5+0j)
```

10) Generating random integers

```
    import random

    x=random.randint(1,100)

print x
```

Output:

```
32
```

11) Using arithmetic operations

```
a=1.6

b=5 c=a+b

print(c)
```

Output:

```
6.6
```

12) Generating floating point integer

```
import random

x=random.uniform(1,100) print(x)
```

Output:

```
92.43244874215033
```

13) Nan, inf and -inf functions

```
import      math

n=math.nan

print(n)



x=float(‘inf’)

x=float(‘-inf’)

print(x) print(y)
```

Output:

```
nan inf

-inf
```

Programs on Array module

14) Create an array in python

```
import array as arr
a=arr.array('i',[1,2,3])
print(a[0])
a.append(5)
print(a)
```

Output:

```
1
array('i', [1, 2, 3, 5])
```

15) Adding elements to an array

```
import array as arr

a=arr.array('i',[1,2,3])

print("integers array before instertion :",*a) a.insert(1,4)

print("integer array before instertion:",*a)
```

Output:

```
integers array before instertion : 1 2 3

integer array before instertion: 1 4 2 3
```

16) Accesing array items

```
    import array as arr
    a=arr.array('i',[1,2,3,4,5,6]
    print(a[0])
print(a[3])
b= arr.array('d',[2.5,3.2,3.3])
print(b[1])
print(b[2])
```

Output:

```
1
4
3.2
3.3
```

17) Removing elements from array

```
import array arr=array.array('i',[1,2,3,1,5])

arr.remove(1)

print(arr)

arr.pop(2)

print(arr)
```

Output:

```
array('i', [2, 3, 1, 5])

array('i', [2, 3, 5])
```

18) Slicing an array

```
    import array as arr

    l=[1,2,3,4,5,6,7,8,9,10]

a=arr.array('i',l) sliced_array=a[3:8]

print(sliced_array) sliced_array=a[5:]

print(sliced_array) sliced_array=a[:]

print(sliced_array)
```

Output:

```
array('i', [4, 5, 6, 7, 8])

array('i', [6, 7, 8, 9, 10])

array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

19) Searching element in an array

```
import array

arr=array.array('i',[1,2,3,1,2,5])

print(arr.index(2))  print(arr.index(1))
```

Output:

```
1

0
```

20) Updating elements in an array

```
import array

arr=array.array('i',[1,2,3,1,2,5])

arr[2]=6

print(arr)

arr[4]=8

print(arr)
```

Output:

```
array('i', [1, 2, 6, 1, 2, 5])

array('i', [1, 2, 6, 1, 8, 5])
```

21) Counting elements in array

```
    import array

    arr=array.array('i',[1,2,3,4,2,5,2])

    count=arr.count(2)

print("Numbers of ocuurance of 2:",count)
```

Output:

```
Numbers of ocuurance of 2: 3
```

22) Reversing elements in an array

```
import array arr=array.array('i',[1,2,3,4,5])

arr.reverse()

print("reversed array:",*arr)
```

Output:

```
reversed array: 5 4 3 2 1
```

23) Extend elements from array

```
    import array as arr

    a=arr.array('i',[1,2,3,4,5])

a.extend([6,7,8,9,10])

print(a)
```

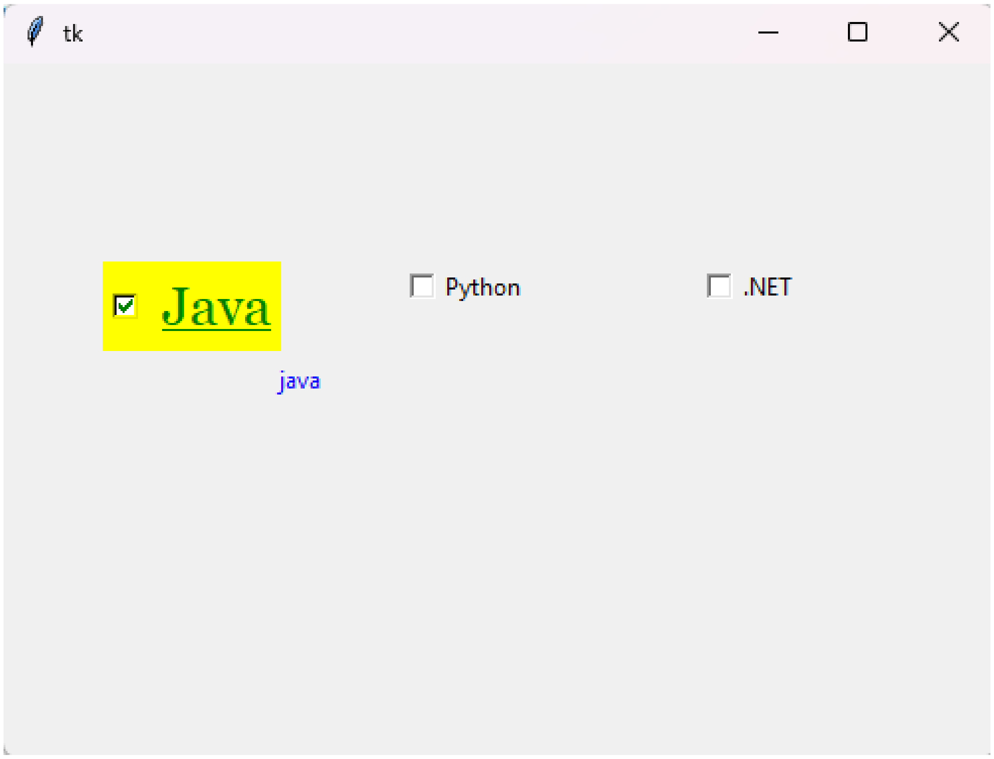
Output:

```
array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

24)Program to display checkbox

```
from tkinter import * class

Mycheck:

    def __init__(self, root):

        self.f =  Frame(root, height=350, width=500)

        self.f.propagate(0)

        self.f.pack()

        self.var1 = IntVar()

        self.var2 = IntVar()

        self.var3 = IntVar()

      self.c1 = Checkbutton(self.f , bg='yellow' , fg = 'green' , font=('Georgia' , 20, 'underline'),
text='Java', variable= self.var1, command=self.display)

        self.c2 = Checkbutton(self.f , text='Python', variable= self.var2, command=self.display)

        self.c3 = Checkbutton(self.f , text='.NET', variable= self.var3, command=self.display)

        self.c1.place(x=50, y=100)

        self.c2.place(x=200, y=100)

        self.c3.place(x=350, y=100)

    def display(self):

        x = self.var1.get()

        y = self.var2.get()

        z = self.var3.get()

        str = ''

        if x==1:

            str += 'java'

        if y==1:
            str+= 'python'
        if z==1:
            str+= '.NET'
    lbl = Label(text=str, fg='blue').place(x=50, y=150, width=200, height=20) root =
Tk()
mb = Mycheck(root) root.mainloop()
```

Output:

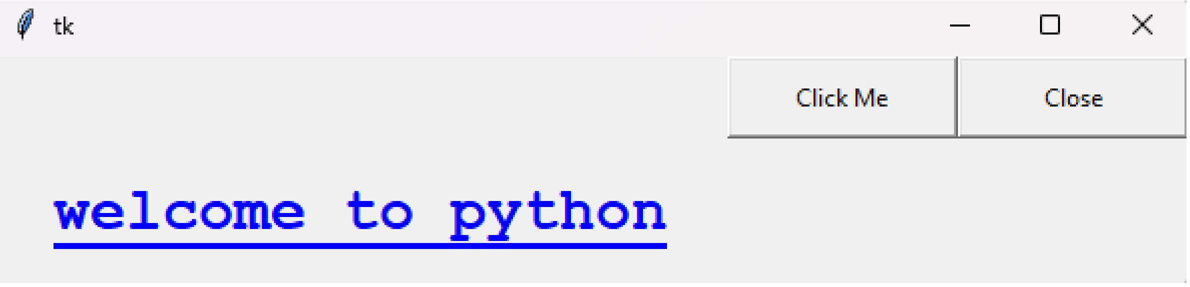


java

25) Program to display Label

```
from tkinter import * class

MyButtons:

    def __init__(self, root):

        self.f = Frame(root, height=350, width=500)

        self.f.propagate(0)

        self.f.pack()

        self.b1 = Button(self.f, text='Click Me', width=15, height=2, command=self.buttonClick)

        self.b2 = Button(self.f, text='Close', width=15, height=2, command=quit)

        self.b1.grid(row=0, column=1)

        self.b2.grid(row=0, column=2)

    def buttonClick(self):

   self.lbl = Label(self.f, text="welcome to python" , width=20, height=2, font=('Courier', -30,
'bold underline '),fg='blue')

   self.lbl.grid(row=2, column=0) root

= Tk()

mb=MyButtons(root)

root.mainloop()
```

Output:

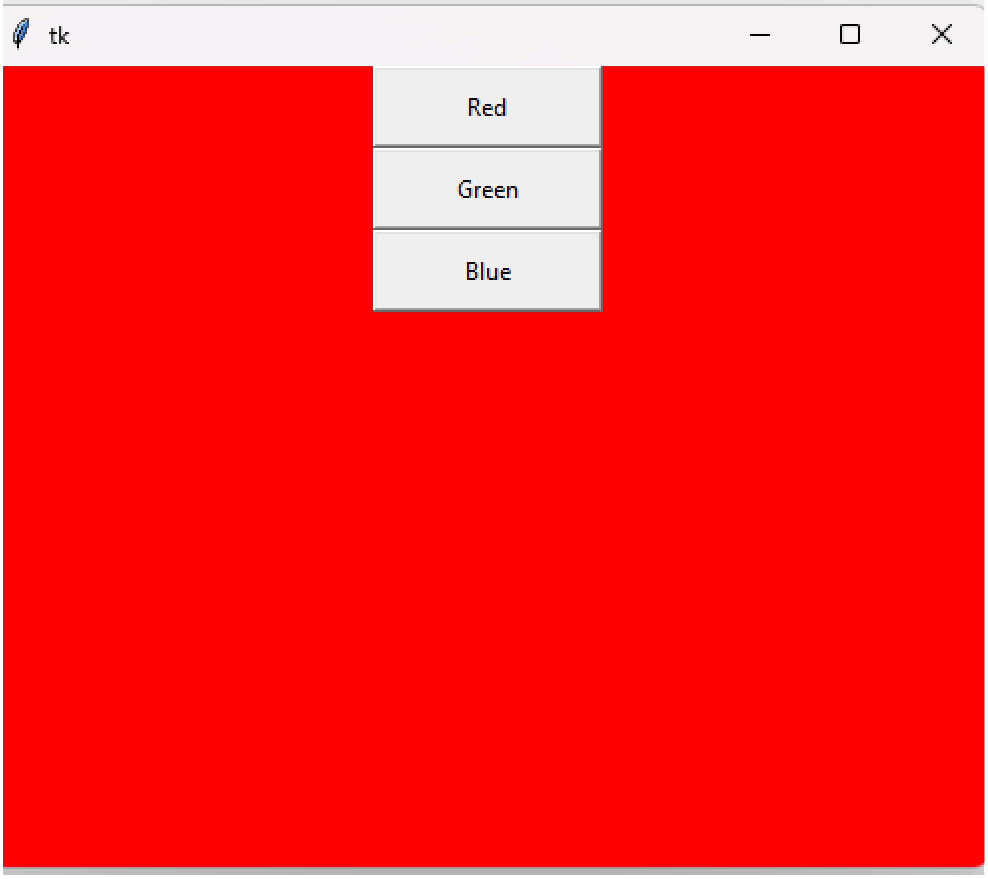


welcome to python

26) Program to display buttons

```
from tkinter import * class

MyButton:

        def __init__(self,root):

           self.f = Frame(root, height=400 , width=500)

           self.f.propagate(0)

           self.f.pack()

      self.b1 = Button(self.f, text='Red' , width=15 , height=2,command=lambda: self.buttonClick(1))

      self.b2 = Button(self.f , text='Green' , width=15, height=2,command=lambda:
self.buttonClick(2))

      self.b3 = Button(self.f , text='Blue' , width=15, height=2,command=lambda:
self.buttonClick(3))

           self.b1.pack()

           self.b2.pack()

           self.b3.pack()

              def buttonClick(self,num):

           if num==1:

              self.f["bg"]= 'red'

           if num==2:

              self.f["bg"]= 'green'

           if num==3:

          self.f["bg"]= 'blue'

root=Tk() mb=MyButton(root)

root.mainloop()
```

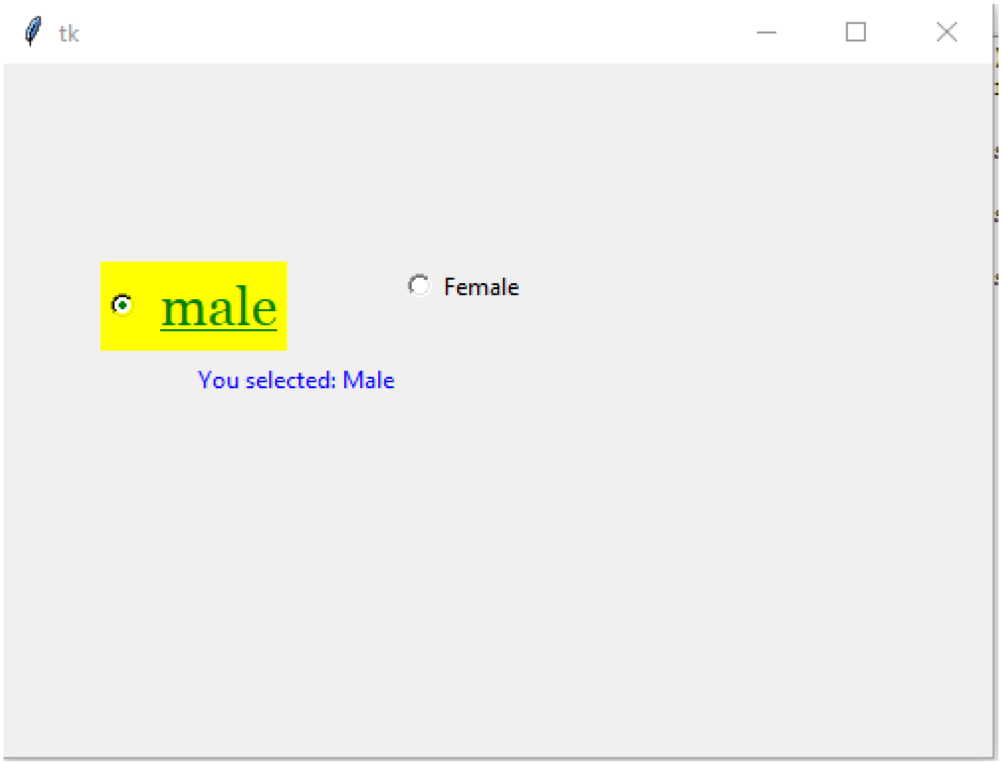
Output:



27) Program to display radiobuttons

```
from tkinter import * class
Myradio:
    def __init__(self , root):
      self.f = Frame(root, height=350, width=500)
      self.f.propagate(0)
      self.f.pack()
      self.var = IntVar()
      self.r1 = Radiobutton(self.f, bg='yellow', fg= 'green' ,
              font=('georgia' , 20, 'underline') , text='male', variable=self.var, value=1,
command=self.display)
   self.r2 = Radiobutton(self.f, text='Female' , variable=self.var, value=2, command=self.display)
      self.r1.place(x=50, y=100)
      self.r2.place(x=200, y=100)
    def display(self):
      x = self.var.get()
      str = "
      if x==1:
        str += 'You selected: Male '
      if x==2:
        str += 'You selected: Female '
   lbl= Label(text=str, fg='blue').place(x=50, y=150, width=200, height=20) root =
Tk()
mb= Myradio(root) root.mainloop()
```

Output:

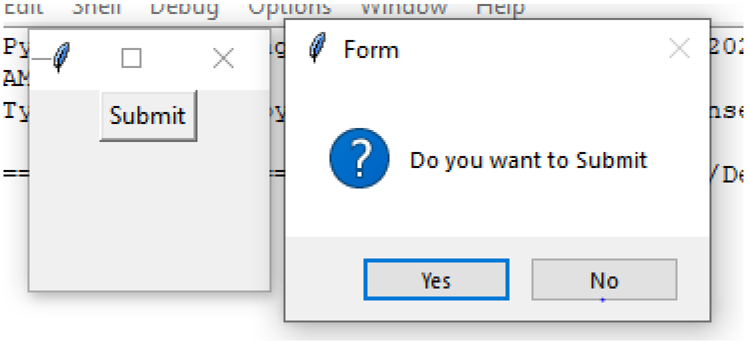


You selected: Male

28) Program to display Dialouge box:

```
from tkinter import *

from tkinter import messagebox

main = Tk()

def Submit():

    messagebox.askquestion("Form", "Do you want to Submit")

main.geometry("100x100")

B1 = Button(main, text="Submit", command=Submit)

B1.pack()

main.mainloop()
```

Output:



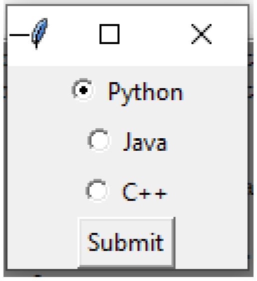
Form

Do you want to Submit

29) Program to display Radio Button 2 :

```
import tkinter as tk

from tkinter import messagebox def

show_choice():

    choice = language.get()

    if choice:

        messagebox.showinfo("Choice", f"You chose: {choice}") else:

   messagebox.showwarning("Choice", "Please select a language.") root =

tk.Tk()

root.title("Language Choice")

language = tk.StringVar()

for lang in ["Python", "Java", "C++"]:

  tk.Radiobutton(root, text=lang, variable=language, value=lang).pack()

tk.Button(root, text="Submit", command=show_choice).pack()

root.mainloop()
```

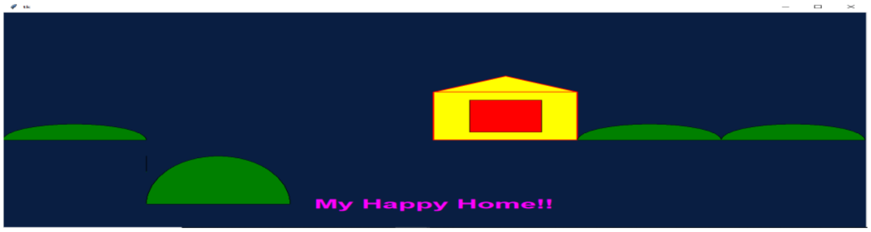
Output :



30) Program to display creation of home

```
from tkinter import *

root=Tk()

c=Canvas(root, bg="#091e42", height=700, width=1200)

c.create_polygon(600,250,700,200,800,250,800,400,600,400,width=2, fill="yellow", outline="red")

c.create_line(600,250,800, 250, width=2, fill="red")

c.create_rectangle(650,275,750,375, fill="red")

x1,y1=0,350

x2,y2=200,450

for i in range(3):

        c.create_arc(x1,y1,x2,y2, start=0, extent=180, fill="green")

        x1+=200

        y1+=200

c.create_arc(800,350,1000,450, start=0, extent=180, fill="green") c.create_arc(1000,350,1200,450,

start=0, extent=180, fill="green")

id=c.create_text(600,600, text="My Happy Home!!", font=('Helvetica', 30, 'bold'), fill="magenta")

c.pack()

root.mainloop
```

Output:

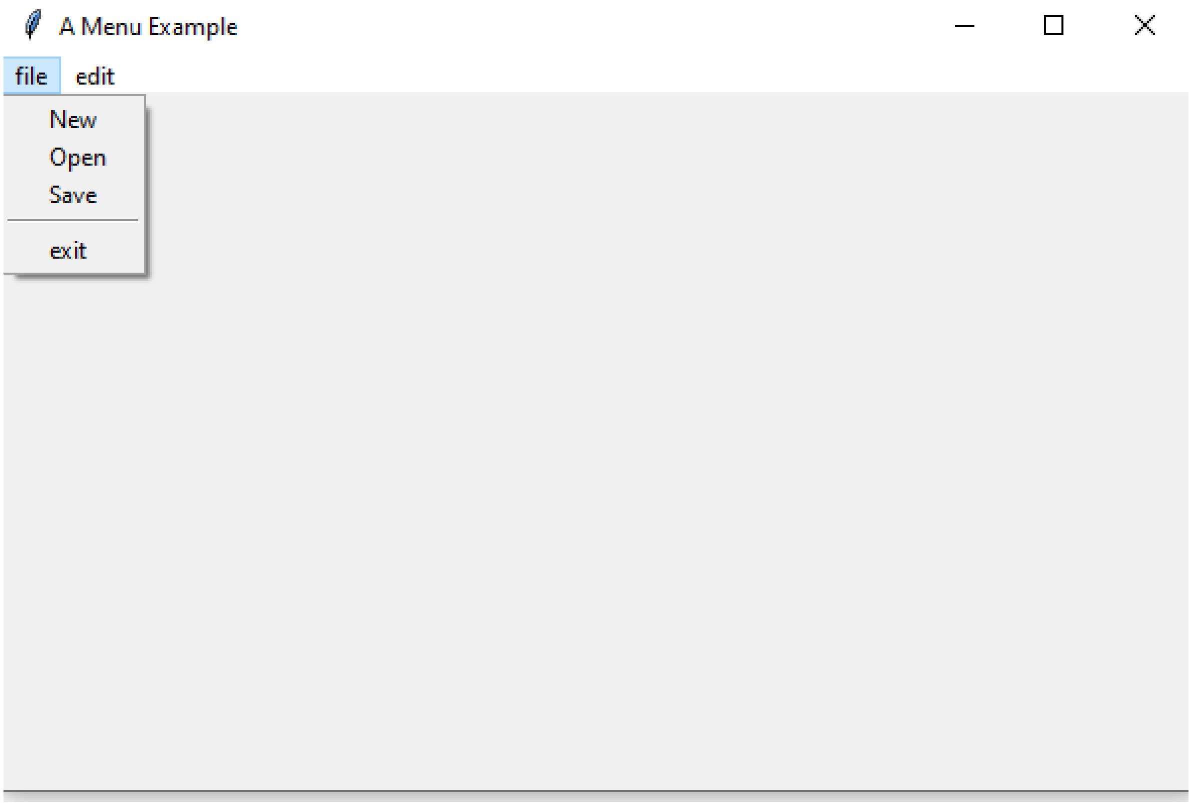


My Happy Home!!

31) Program to display menu

```
from tkinter import *

class MyMenuDemo:

    def __init__(self, root):

        self.menubar = Menu(root)

        root.config(menu=self.menubar)

        self.filemenu= Menu(root, tearoff=0)

        self.filemenu.add_command(label="New", command=self.donothing)

        self.filemenu.add_command(label="Open", command=self.donothing)

        self.filemenu.add_command(label="Save", command=self.donothing)

        self.filemenu.add_separator()

        self.filemenu.add_command(label="exit", command=root.destroy)

        self.menubar.add_cascade(label="file", menu=self.filemenu)

        self.editmenu = Menu(root, tearoff=0)

        self.editmenu.add_command(label='cut', command=self.donothing)

        self.editmenu.add_command(label='copy', command=self.donothing)

        self.editmenu.add_command(label='paste', command=self.donothing)

        self.menubar.add_cascade(label="edit", menu=self.editmenu)

    def donothing(self):

       pass

root = Tk()

root.title("A Menu Example") obj=

MyMenuDemo(root)

root.geometry('600x350')

root.mainloop()
```

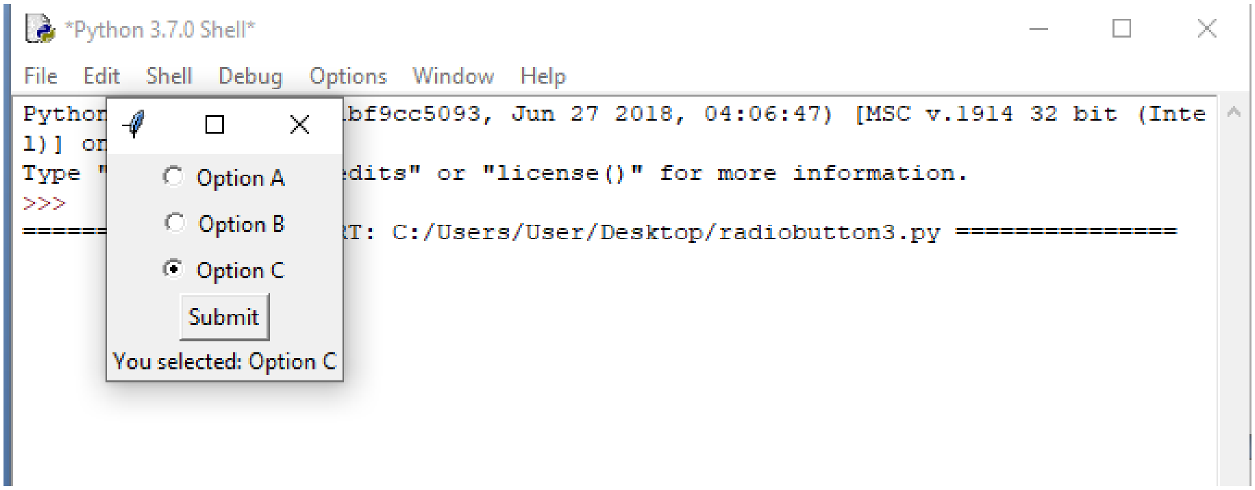
Output:



32) Write a program using radio buttons in python and application should contain submit button after selecting the radio buttons

```
import tkinter as tk
def submit_selection():
    if choice.get() == 1:
        selected_option = "Option A"
    elif choice.get() == 2:
        selected_option = "Option B"
    elif choice.get() == 3:
        selected_option = "Option C"
    else:
        selected_option = "No option selected"
    result_label.config(text=f"You selected: {selected_option}")
window = tk.Tk()
window.title("Radio Button Example")
choice = tk.IntVar()
radio_button1 = tk.Radiobutton(window, text="Option A", variable=choice, value=1)
radio_button2 = tk.Radiobutton(window, text="Option B", variable=choice, value=2)
radio_button3 = tk.Radiobutton(window, text="Option C", variable=choice, value=3)
radio_button1.pack()
radio_button2.pack()
radio_button3.pack()
submit_button = tk.Button(window, text="Submit", command=submit_selection)
submit_button.pack()
result_label = tk.Label(window, text="")
result_label.pack()
window.mainloop()
```

Output:



```
Pythor          bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
1)] or
Type "          edits" or "license()" for more information.
>>>
======          T: C:/Users/User/Desktop/radiobutton3.py ===============
```

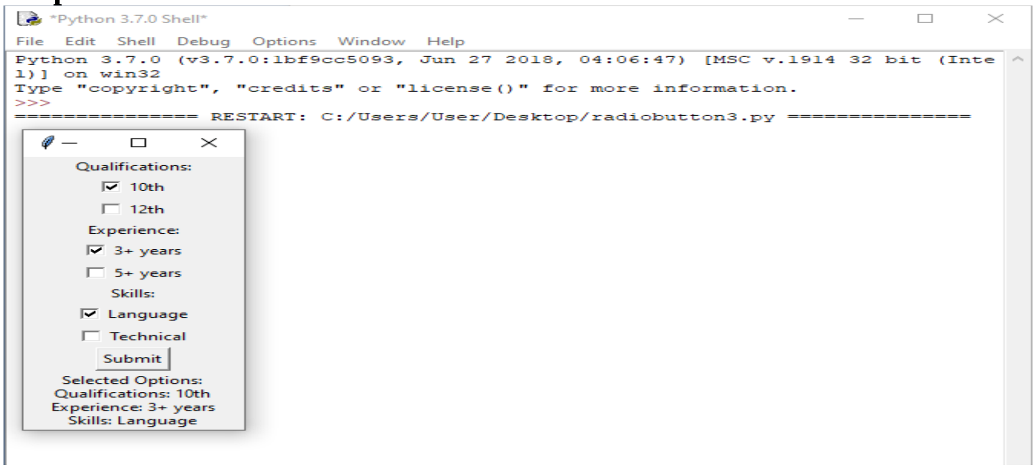
You selected: Option C

33)Write a program using check box in python to display any 3 categories after selecting the options click on submit button

```
import tkinter as tk
def submit_selections():
    selected_qualifications = []
    if qualification_10th.get():
        selected_qualifications.append("10th")
    if qualification_12th.get():
        selected_qualifications.append("12th")
    selected_experiences = []
    if experience_3plus.get():
        selected_experiences.append("3+ years")
    if experience_5plus.get():
        selected_experiences.append("5+ years")
    selected_skills = []
    if skills_language.get():
        selected_skills.append("Language")
    if skills_technical.get():
        selected_skills.append("Technical")
    result_text = "Selected Options:\n"
    if selected_qualifications:
        result_text += "Qualifications: " + ", ".join(selected_qualifications) + "\n"
    if selected_experiences:
        result_text += "Experience: " + ", ".join(selected_experiences) + "\n"
    if selected_skills:
     result_text += "Skills: " + ", ".join(selected_skills)
    result_label.config(text=result_text)
window = tk.Tk()
window.title("Checkbox Example")
qualification_frame = tk.Frame(window)
qualification_frame.pack()
qualification_label = tk.Label(qualification_frame, text="Qualifications:")
qualification_label.pack()
qualification_10th = tk.IntVar()
qualification_12th = tk.IntVar()
qualification_checkbox_10th = tk.Checkbutton(qualification_frame, text="10th",
variable=qualification_10th)
qualification_checkbox_12th = tk.Checkbutton(qualification_frame, text="12th",
variable=qualification_12th)
qualification_checkbox_10th.pack()
qualification_checkbox_12th.pack()
experience_frame = tk.Frame(window)
experience_frame.pack()
experience_label = tk.Label(experience_frame, text="Experience:")
experience_label.pack()
experience_3plus = tk.IntVar()
experience_5plus = tk.IntVar()
experience_checkbox_3plus = tk.Checkbutton(experience_frame, text="3+ years",
variable=experience_3plus)
experience_checkbox_5plus = tk.Checkbutton(experience_frame, text="5+ years",
variable=experience_5plus)
```

```
experience_checkbox_3plus.pack()
experience_checkbox_5plus.pack()
skills_frame = tk.Frame(window)
skills_frame.pack()
skills_label = tk.Label(skills_frame, text="Skills:")
skills_label.pack()
skills_language = tk.IntVar()
skills_technical = tk.IntVar()
skills_checkbox_language = tk.Checkbutton(skills_frame, text="Language",
variable=skills_language)
skills_checkbox_technical = tk.Checkbutton(skills_frame, text="Technical",
variable=skills_technical)
skills_checkbox_language.pack()
skills_checkbox_technical.pack()
submit_button = tk.Button(window, text="Submit", command=submit_selections)
submit_button.pack()
result_label = tk.Label(window, text="")
result_label.pack()
window.mainloop()
```

Output:



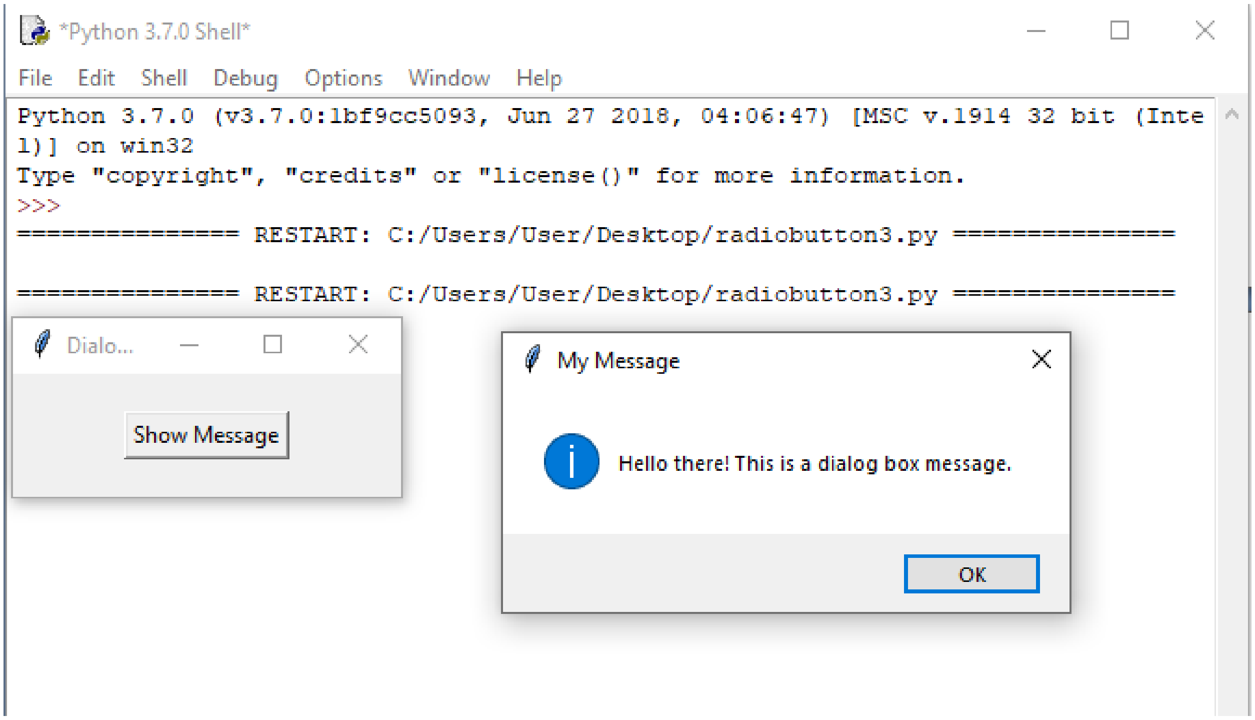
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
1)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/User/Desktop/radiobutton3.py ===============
```

Qualifications:
☑ 10th
☐ 12th
Experience:
☑ 3+ years
☐ 5+ years
Skills:
☑ Language
☐ Technical
Submit
Selected Options:
Qualifications: 10th
Experience: 3+ years
Skills: Language

34)write a program on your own to display the message using the dialog box

```
import tkinter as tk
from tkinter import messagebox
def show_message():
   messagebox.showinfo("My Message", "Hello there! This is a dialog box message.")
window = tk.Tk()
window.title("Dialog Box Example")
message_button = tk.Button(window, text="Show Message", command=show_message)
message_button.pack(pady=20)
window.mainloop()
```

Output:



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/User/Desktop/radiobutton3.py ===============

=============== RESTART: C:/Users/User/Desktop/radiobutton3.py ===============
```

Hello there! This is a dialog box message.

35) Program using arrays to display the students grades

```
def display_grades(names, grades):
    print("\nStudent Grades:")
    print("-------------------")
    for i in range(len(names)):
        print(f"{names[i]}: {grades[i]}")

def main():
        student_names = ['Alice', 'Bob', 'Charlie', 'Diana', 'Eve']

      student_grades = ['A', 'B', 'C', 'A', 'B']

        display_grades(student_names, student_grades)

if __name__ == "__main__":
    main()
```

OUTPUT:

```
Student Grades:
-------------------
Alice: A
Bob: B
Charlie: C
Diana: A
Eve: B
```

36) Program using arrays to find whether the student has passed or failed

```
def check_pass_or_fail(names, grades):
    print("\nStudent Status:")
    print("-------------------")

    for i in range(len(names)):
               if grades[i] in ['A', 'B', 'C']:
            status = 'Passed'
        else:
            status = 'Failed'

        print(f"{names[i]}: {status}")

def main():
        student_names = ['Alice', 'Bob', 'Charlie', 'Diana', 'Eve']

        student_grades = ['A', 'D', 'C', 'B', 'F']  # Example grades

       check_pass_or_fail(student_names, student_grades)

if __name__ == "__main__":
    main()
```

OUTPUT:

```
Student Status:
-------------------
Alice: Passed
Bob: Failed
Charlie: Passed
Diana: Passed
Eve: Failed
```

37) Program using arrays to display the number of Books, Publisher name and Year of Publication

```
def display_book_details(book_titles, publishers, years):
    print("\nBook Details:")
    print("------------------------------")
    for i in range(len(book_titles)):
        print(f"Book: {book_titles[i]}")
        print(f"Publisher: {publishers[i]}")
        print(f"Year of Publication: {years[i]}")
        print("------------------------------")

def main():

       book_titles = ['The Great Gatsby', '1984', 'To Kill a Mockingbird']
    publishers = ['Scribner', 'Secker & Warburg', 'J.B. Lippincott & Co.']
    years = [1925, 1949, 1960]

      display_book_details(book_titles, publishers, years)

if __name__ == "__main__":
    main()
```

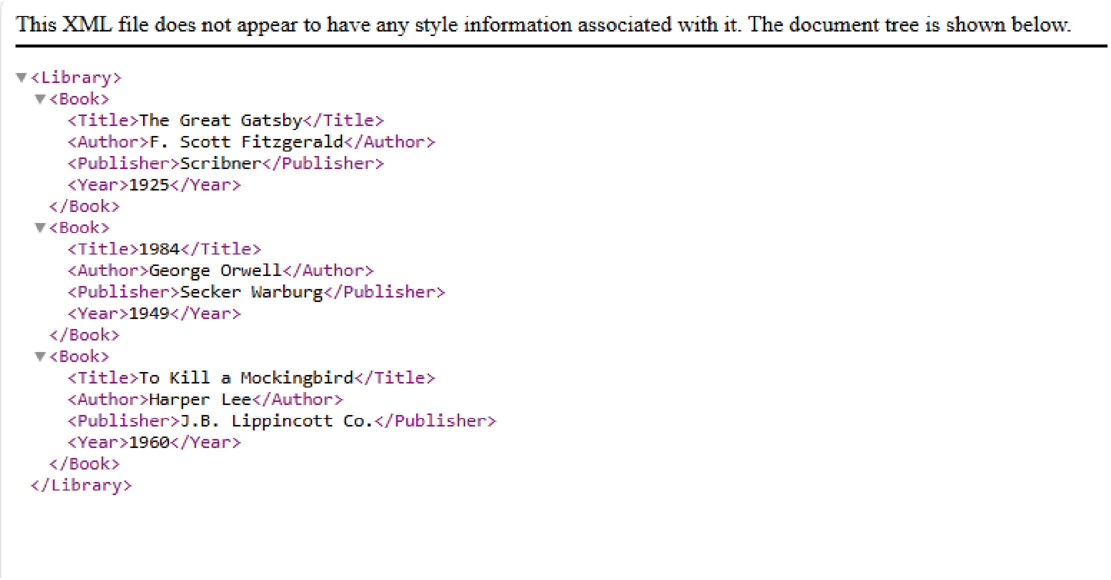
OUTPUT:

```
Book Details:
------------------------------
Book: The Great Gatsby
Publisher: Scribner
Year of Publication: 1925
------------------------------
Book: 1984
Publisher: Secker & Warburg
Year of Publication: 1949
------------------------------
Book: To Kill a Mockingbird
Publisher: J.B. Lippincott & Co.
Year of Publication: 1960
------------------------------
```

38) XML program to create the library information

```
<?xml version="1.0" encoding="UTF-8"?>
<Library>
   <Book>
      <Title>The Great Gatsby</Title>
      <Author>F. Scott Fitzgerald</Author>
      <Publisher>Scribner</Publisher>
      <Year>1925</Year>
   </Book>
   <Book>
      <Title>1984</Title>
      <Author>George Orwell</Author>
      <Publisher>Secker & Warburg</Publisher>
      <Year>1949</Year>
   </Book>
   <Book>
      <Title>To Kill a Mockingbird</Title>
      <Author>Harper Lee</Author>
      <Publisher>J.B. Lippincott & Co.</Publisher>
      <Year>1960</Year>
   </Book>
</Library>
```

OUTPUT:



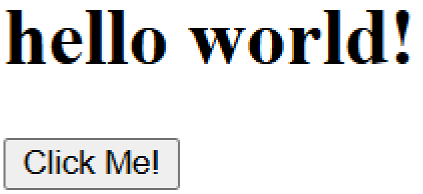
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Library>
  ▼<Book>
     <Title>The Great Gatsby</Title>
     <Author>F. Scott Fitzgerald</Author>
     <Publisher>Scribner</Publisher>
     <Year>1925</Year>
   </Book>
  ▼<Book>
     <Title>1984</Title>
     <Author>George Orwell</Author>
     <Publisher>Secker Warburg</Publisher>
     <Year>1949</Year>
   </Book>
  ▼<Book>
     <Title>To Kill a Mockingbird</Title>
     <Author>Harper Lee</Author>
     <Publisher>J.B. Lippincott Co.</Publisher>
     <Year>1960</Year>
   </Book>
 </Library>
```

39) Program to demonstrate DOM

```
<!DOCTYPE html>
<html>
<body>
<h1 id="demo">this is a heading</h1>
<button type="button"
 onclick="document.getElementById('demo' ).innerHTML='hello world!' ">Click Me!
</button>
</body>
</html>
```

Output:

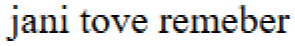


hello world!

40) Program to save the xml file as html file

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<from>jani</from>
<to>tove</to>
<message>remeber</message>
</note>
```

Output:



```
jani tove remeber
```