HW1 Solution

Step 1

```
import pygal
def fibonacci(Nterms: int, N1: int, N2: int):
    Flist= []
    Count = 0
    # check if the number of terms is valid
    if Nterms <= 0:
        print("Please enter a positive integer")
    # if there is only one term, return n1
    elif Nterms == 1:
        print("Fibonacci sequence upto",Nterms,":")
        print(N1)
    # generate fibonacci sequence
    else:
        print("Fibonacci sequence:")
    while Count < Nterms:
        print(N1)
        Nth = N1 + N2
        # update values
        N1 = N2
        N2 = Nth
        Count += 1
        Flist.append(N1)
    return Flist
# nterms = int(input("How many terms? "))

nterms = 10
Flist = fibonacci(nterms, 0, 1)
bar_chart = pygal.Bar() # Then create a bar graph object
bar_chart.add('Fibonacci', Flist) # Add values
bar_chart.render_to_file("hw1Step1-10.svg")

nterms = 50
Flist = fibonacci(nterms, 0, 1)
bar_chart = pygal.Bar() # Then create a bar graph object
bar_chart.add('Fibonacci', Flist)# Add values
bar_chart.render_to_file("hw1Step1-50.svg")
```

Step 2

```
import plotly.graph_objects as go

def fibonacci(Nterms: int, N1: int, N2: int):
      Flist= []
      Count = 0
      # check if the number of terms is valid
      if Nterms <= 0:
            print("Please enter a positive integer")
      # if there is only one term, return n1
      elif Nterms == 1:
            print("Fibonacci sequence upto",Nterms,":")
            print(N1)
      # generate fibonacci sequence
      else:
            print("Fibonacci sequence:")
      while Count < Nterms:
            print(N1)
            Nth = N1 + N2
            # update values
            N1 = N2
            N2 = Nth
            Count += 1
            Flist.append(N1)
      return Flist

# nterms = int(input("How many terms? "))

nterms = 10
Flist = fibonacci(nterms, 0, 1)
fig = go.Figure(data=go.Bar(y=Flist))
fig.write_html('10bars_figure.html', auto_open=True)

nterms = 50
Flist = fibonacci(nterms, 0, 1)
fig = go.Figure(data=go.Bar(y=Flist))
fig.write_html('50bars_figure.html', auto_open=True)
```

Step 3

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import pandas as pd

nterms = 10
Flist10 = fibonacci(nterms, 0, 1)
barnumber10 = list(range(len(Flist10)))
nterms = 50
Flist50 = fibonacci(nterms, 0, 1)
barnumber50 = list(range(len(Flist50)))
external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
df10 = pd.DataFrame({
      "Value": barnumber10,
      "Amount": Flist10
})
df50 = pd.DataFrame({
      "Value": barnumber50,
      "Amount": Flist50
})
fig10 = px.bar(df10, x="Value", y="Amount")
fig50 = px.bar(df50, x="Value", y="Amount")
app.layout = html.Div(children=[
      html.H1(children='Homework 1'),
      dcc.Graph(
            id='Figure1',
            figure=fig10,
            # figure=fig50,
      )
]),
html.Div(children=[
      dcc.Graph(
            id='Figure2',
            # figure=fig10,
            figure=fig50,
      )
]),
if __name__ == '__main__':
      app.run_server(debug=False)
```