In [2]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
```

In [3]:
```
X = 100
sigma = 0.5
T = 1
Delta_t = T/365
r = 0.01
r_period = (1+r)**Delta_t-1
N=365
```

In [5]:
```
up = np.exp(sigma * np.sqrt(Delta_t))
dn = 1/up
Aup = (1+r_period-dn)/((1+r_period)*(up-dn))
Adn = (up-1-r_period)/((1+r_period)*(up-dn))
print(Aup)
print(Adn)
```

```
0.4939648799091735
0.5060078592819895
```

In [7]:
```
print(np.zeros((N+1,N+1)))
np.zeros((N+1,N+1)).size
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Out[7]: 133956

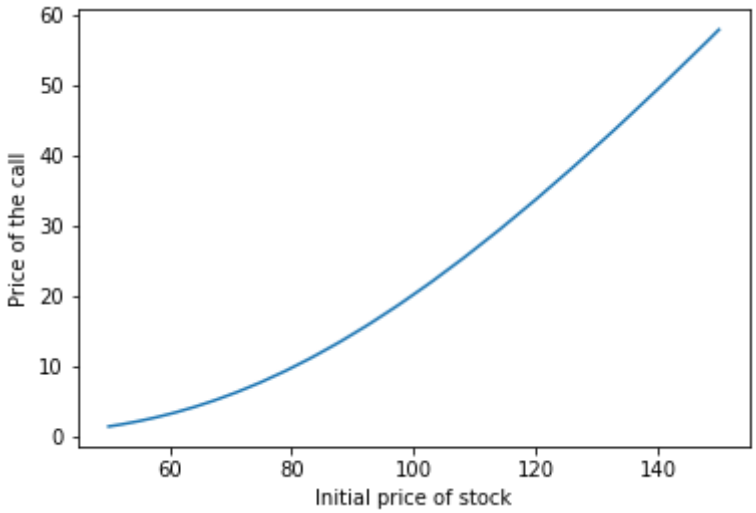
In [8]:
```
def Bin_call(S0,X,delta_t,r_period):
    option = np.zeros((N+1,N+1))
    for i in range(N+1):
        option[i,N] = max(0,S0*up**(i)*dn**(N-i)-X)
    for j in range(N-1,-1,-1):
        for k in range(j+1):
            option[k,j] = Aup*option[k+1,j+1] + Adn*option[k,j+1]
    return(option[0,0])
```

In [11]:
```
s0 = list(range(50,151))

Call_Prices = []
for i in range(len(s0)):
    Call_Prices.append(Bin_call(s0[i],X,Delta_t,r_period))
print(Call_Prices)
```

```
[1.3579363900822998, 1.495524225126931, 1.6335599808539547, 1.795523900130
7461, 1.9574878194075391, 2.125533678230216, 2.314520476762168, 2.503507
2752941194, 2.70083954675241, 2.9194565110762927, 3.138073475400176, 3.36
24770783196967, 3.6132287585806813, 3.8639804388416654, 4.11473211910264
8, 4.396915484139703, 4.682138302588601, 4.967361121037502, 5.26969918284
3048, 5.591496646318429, 5.913294109793817, 6.235091573269202, 6.58909397
5462142, 6.94927543894717, 7.30945690243219, 7.670239214667525, 8.0702651
38244936, 8.470291061822344, 8.87031698539975, 9.270342908977163, 9.71069
248021062, 10.151628921093467, 10.592565361976323, 11.033501802859181, 1
1.505171593617714, 11.987656283382112, 12.470140973146508, 12.95262566291
0908, 13.446195228146134, 13.970417024780367, 14.4946388214146, 15.018860
618048834, 15.54308241468307, 16.09015682481769, 16.655849637205428, 17.2
21542449593173, 17.78723526198091, 18.352928074368634, 18.94239012967373,
19.54884163847792, 20.155293147282116, 20.761744656086314, 21.36819616489
0506, 21.98844974269238, 22.634524443902354, 23.280599145112337, 23.92667
384632233, 24.572748547532314, 25.218823248742307, 25.89660509466592, 26.
580780430196604, 27.264955765727283, 27.94913110125799, 28.6333064367886
7, 29.31819343627241, 30.038607106380386, 30.759020776488327, 31.47943444
659627, 32.19984811670423, 32.9202617868122, 33.640675456920135, 34.39159
277948424, 35.14609880200263, 35.90060482452101, 36.65511084703938, 37.40
961686955779, 38.16412289207617, 38.93261007924654, 39.71884080522511, 4
0.50507153120373, 41.29130225718234, 42.07753298316093, 42.8637637091395
3, 43.64999443511812, 44.454909118524995, 45.27034025132889, 46.085771384
132755, 46.901202516936635, 47.7166336497405, 48.53206478254435, 49.34749
591534822, 50.17551613710002, 51.01753183390902, 51.85954753071806, 52.70
1563227527046, 53.54357892433607, 54.38559462114513, 55.22761031795412, 5
6.069626014763145, 56.933717733939304, 57.799673080203796]
```

In [12]:
```
plt.plot(s0,Call_Prices)
plt.xlabel("Initial price of stock")
plt.ylabel("Price of the call")
plt.show()
```

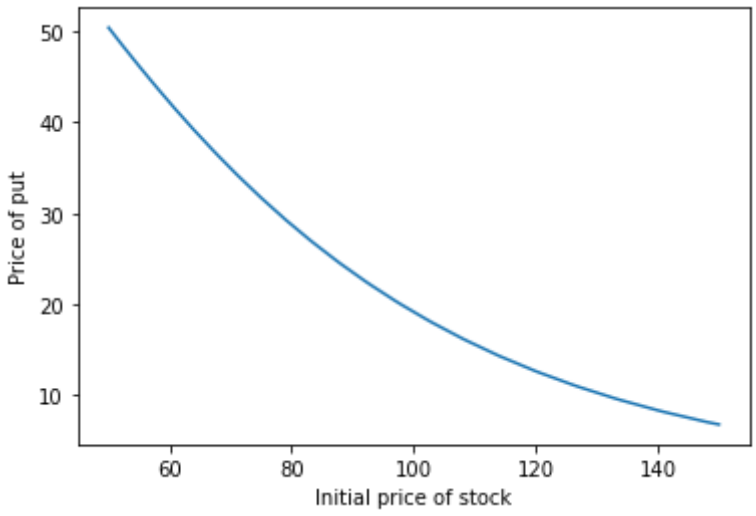


In [16]:
```
def Bin_Put(S0,X,delta_t,r_period):
    option = np.zeros((N+1,N+1))
    for i in range(N+1):
        option[i,N] = max(0,X-S0*up**(i)*dn**(N-i))
    for j in range(N-1,-1,-1):
        for k in range(j+1):
            option[k,j] = Aup*option[k+1,j+1] + Adn*option[k,j+1]
    return(option[0,0])
```

In [17]:
```
Put_Prices = []
for i in range(len(s0)):
    Put_Prices.append(Bin_Put(s0[i],X,Delta_t,r_period))
    print(Put_Prices)
```

```
26889644, 14.328094426373385]
```

In [18]:
```
plt.plot(s0,Put_Prices)
plt.xlabel("Initial price of stock")
plt.ylabel("Price of put")
plt.show()
```



In [20]:
```
PV = [X*np.exp(-r_period*365)]*len(s0)

Parity = np.add(np.subtract(Call_Prices,Put_Prices),np.subtract(PV,s0))
print(Parity)
```

```
[-1.34287235e-05 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05
 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05
 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05
 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05
 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05 -1.34287235e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287235e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05 -1.34287234e-05
 -1.34287234e-05 -1.34287234e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05 -1.34287233e-05
 -1.34287233e-05 -1.34287232e-05 -1.34287233e-05 -1.34287232e-05
 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05
 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05
 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05
 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05 -1.34287232e-05
 -1.34287232e-05]
```

In [24]:
```
S0 = 100
FV = 100
sigma = 0.5
T = 1
Delta_t = T/365
r = 0.01
r_period = (1+r)**Delta_t-1
N=365

up = np.exp(sigma * np.sqrt(Delta_t))
dn = 1/up
Aup = (1+r_period-dn)/((1+r_period)*(up-dn))
Adn = (up-1-r_period)/((1+r_period)*(up-dn))

bin = np.zeros((184,184))
for i in range(184):
    bin[i,183] = S0*(up**(i))*(dn**(183-i))

bond = bin
bond[:,183][bond[:,183] <= FV*np.exp(-183*r_period)] = FV*np.exp(-183*r_per

for j in range(182,-1,-1):
    for k in range(j+1):
        bond[k,j] = Aup*bond[k+1,j+1] + Adn*bond[k,j+1]

bond_0 = bond[0,0]
print("The value of the convertible ZCB at t=0 is",bond_0)

option_0 = bond[0,0]-FV*np.exp(-0.01)
print("The value of the option to convert is", option_0)
```

```
The value of the convertible ZCB at t=0 is 113.50731355104767
The value of the option to convert is 14.502330176130869
```