CAPTION GENERATOR

In [ ]:
```
import numpy as np
import string
from tkinter import *
from tkinter import filedialog
import PIL.Image
import PIL.ImageTk
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.preprocessing.image import load_img, img_to_array
from keras.models import load_model
from pickle import load
from IPython.display import display, Image
```

In [ ]:
```
cnn_model = load_model("CNN_Model.h5")
rnn_model = load_model("RNN_Model.h5")
tokenizer = load(open("Flickr8K_Tokenizer.p", "rb"))
word_to_index = tokenizer.word_index
index_to_word = dict([index, word] for word, index in word_to_index.items())
vocab_size = len(tokenizer.word_index) + 1
max_len = 28
```

In [5]:
```
root = Tk()
root.title("Image Caption Generator")
root.state('zoomed')
root.resizable(width = True, height = True)

panel = Label(root, text = 'IMAGE CAPTION GENERATOR', font = ("Arial", 30))
panel.place(relx = 0.3, rely = 0.1)

filename = None
def chooseImage(event = None):
    global filename
    filename = filedialog.askopenfilename()
    img = PIL.Image.open(filename)
    img = img.resize((350, 300))
    img = PIL.ImageTk.PhotoImage(img)
    display_image = Label(root, image = img)
    display_image.image = img
    display_image.place(relx=0.4,rely=0.2)

value = StringVar()
def generateCaption(event = None):
    if(filename == None):
        value.set("No Image Selected")
    else:
        img = load_img(filename, target_size = (299, 299))
        img = img_to_array(img)
        img = np.expand_dims(img, axis = 0)
        img = img / 127.5
        img = img - 1.0
        features = cnn_model.predict(img)
        in_text = 'startseq'
        max_words = 9
        words_generated = 0
        for i in range(max_len):
            sequence = tokenizer.texts_to_sequences([in_text])[0]
            sequence = pad_sequences([sequence], maxlen=max_len)  # Pad sequenc
            pred = rnn_model.predict([features,sequence], verbose=0)
            pred = np.argmax(pred)
            word = index_to_word[pred]
            if word is None:
                break
            in_text += ' ' + word
            words_generated += 1
            if word == 'endseq' or words_generated >= max_words:
                break
        in_text = ' '.join(in_text.split()[1: -1])
        in_text = in_text[0].upper() + in_text[1:] + '.'
        value.set(in_text)
    #display_caption = Label(root, textvariable = value, font=("Arial",18))
    #display_caption.place(relx = 0.48, rely = 0.85)

    display(Image(filename))
    print(in_text)

button1 = Button(root, text='Choose an Image', font=(None, 18), activeforegrour
button1.place(relx = 0.3, rely = 0.65)
button2 = Button(root, text='Generate Caption', font=(None, 18), activeforegrou
```

```
button2.place(relx = 0.56, rely = 0.65)
caption = Label(root, text='Caption : ', font=("Arial", 18))
caption.place(relx = 0.35, rely = 0.85)

root.mainloop()
```

```
WARNING:tensorflow:No training configuration found in the save file, so the m
odel was *not* compiled. Compile it manually.
1/1 [==============================] - 2s 2s/step
```

```
Climber is standing on the top of mountain.
1/1 [==============================] - 0s 171ms/step
```

```
Dog with stick in its mouth is running.
```

In [ ]: