

## Dynamic Array

- A Dynamic Array automatically grows when you try to make an insertion and there is no more space left for the new item. Usually, the array double in size

In [1]:

```
import ctypes
```

In [49]:

```

class DynamicArray:
    def __init__(self):
        self.n=0 # count actual elements
        self.size=1 # default array capacity
        self.A=self._make_array(self.size) # Low Level arrays-->creating an array
    def __len__(self):
        return self.n
    def append(self,item):
        if self.n==self.size: # not enough memory or rooms
            self._resize(2*self.size) # double the capacity
        self.A[self.n]=item # I am inserting element into the array
        self.n+=1
    def _resize(self,new_capacity):
        # 1.Creata new Array
        B=self._make_array(new_capacity)
        self.size=new_capacity
        # 2.Copy the elemnts from old array to new array
        for i in range(self.n):
            B[i]=self.A[i]
        #3 .Reassign(delete the previous array)
        self.A=B
    def _make_array(self,new_capacity):
        return (new_capacity*ctypes.py_object)()
    def __str__(self):
        temp=""
        for i in range(self.n):
            temp=temp+str(self.A[i])+","
        temp=temp[:-1]
        return "["+temp+"]"
    def __getitem__(self,index):
        if 0<=index<=self.n:
            return self.A[index] # It will reteive an element from an array
        else:
            return "IndexErrror"
    def __delitem__(self,index):
        if 0<=index<=self.n:
            for i in range(index,self.n-1):
                self.A[i]=self.A[i+1]
            self.n-=1
        else:
            return "IndexError"

arr=DynamicArray() # object creation

```

In [50]:

```
arr.append(100)
```

In [51]:

```
arr.append(500)
```

In [52]:

```
arr.append(200)
```

In [53]:

```
arr.append(300)
```

In [54]:

```
arr.append(20000)
```

In [55]:

```
print(len(arr))
```

5

In [56]:

```
print(arr)
```

[100, 500, 200, 300, 20000]

In [57]:

```
print(arr[1])
```

500

In [58]:

```
del arr[2]
```

In [59]:

```
print(arr)
```

[100, 500, 300, 20000]

In [ ]: