Test Result

直接编译运行main.cpp

注意 Vector里可以存放各种类型的数据，在我的main.cpp开头 `#define TYPE int` ，即代表此时存入Vector的类型为 `int` 型，你可以自己修改TYPE类型使Vector可以放入你想放入的元素的类型，比如 `#define TYPE double` 或者 `#define TYPE char`

根据提示语进行测试：

```
Now let's test the ctor with size: // 有参构造的测试
Please input five elements:
1 2 3 4 5 // 输入五个数据，我在程序里把TYPE设置成了int，所以此时用int测试
After pushing back the Vector:
Vint.empty() = 0
The current elements in Vector are:
1 2 3 4 5
Access the element with index -1(actually it is illegal):
std::out_of_range
Please input the index of the element you want to access:
2 // 输入你想访问的下标
3 // 输出你想访问的下标的值
Vint.size() = 5
After Finishing Vint.clear()
Vint.size() = 0
Vint.empty() = 1
Now let's test the ctor without size: // 无参构造的测试
Please input the size of the element you want to push back:
8 // 输入你想存到Vector里的元素数量，我这里输入了8
1 2 3 4 5 6 7 8 // 输入8个想要pushback到Vector里的元素，元素类型为TYPE
After pushing back the Vector, The current elements in Vector are:
1 2 3 4 5 6 7 8
Now we finish the test, BYE! // 结束测试
```

```
Now let's test the ctor with size:
Please input five elements:
1 2 3 4 5
After pushing back the Vector:
Vint.empty() = 0
The current elements in Vector are:
1 2 3 4 5
Access the element with index -1(actually it is illegal):
std::out_of_range
Please input the index of the element you want to access:
2
3
Vint.size() = 5
After Finishing Vint.clear()
Vint.size() = 0
Vint.empty() = 1
Now let's test the ctor without size:
Please input the size of the element you want to push back:
8
1 2 3 4 5 6 7 8
After pushing back the Vector, The current elements in Vector are:
1 2 3 4 5 6 7 8
Now we finish the test, BYE!
```