

实验一

姓名：黄丹禹

学号：2012030

日期：2022/10/26

一、实验平台

- a) 基于 Visual Studio 2022，使用C++语言编写

b) 调用函数库：

```
#include <iostream>
#include <vector>
#include <string>
```

二、实验目的

在所选环境下定义一个信号序列，要求

- a) 序列长度有限（长度自定，例如 $n=-3:8$ ）
- b) 该序列起始位置不为 0，例如上面 $n=-3$ 开始
- c) 能够读取、写入该序列任意位置例如 $x[n]$, $n=-3:8$, 令 $x[3]=5$; $x[4]=x[4]-2$ 等等
- d) 完成无限长序列的输入（右侧序列-即不停输入直到停止指令）

三、实验核心公式及问题

- a) 定义 seq 类，在类中定义 begin, end 两个变量作为 n 的起始与终止值。定义一个 int 型的 vector 数组用以承载数据。
- b) 主要通过 vector 的 push_back() 函数和循环来实现
- c) 对序列的读取、写入、修改相当于在 seq 类中增加相应功能的函数。
- d) 无限长数组的实现需要提前选定希望执行的操作类型，将输入置于 while 循环中，输入后进行实时处理后可以继续输入，直到程序结束。

四、 实验设计

a) 定义信号序列:

类中成员变量定义:

```
class seq {  
    vector<double> a;  
    int begin;  
    int end;  
};
```

初始化:

① 构造函数:

```
seq(int b, int e, vector<double> temp) {  
    begin = b;  
    end = e;  
    a = temp;  
}
```

②实例化:

```
vector<double> u = {1, 2, 3};  
seq A(0, 2, u);  
vector<double> u2 = {1, 2, 3};  
seq B(-2, 0, u2);
```

b) 读取信号数据值:

Index (即时间 n 的值) 与 begin 的差就是 vector 中的索引值:

```
//通过n获取数据  
int seq::getNum(int index) {  
    return this->a[index - this->begin];  
}
```

测试数据: A={1, 2, 3} , n=0:2。获取 n=1 时的数据:

```
A.getNum(1):2
```

c) 更改信号数据值:

原理与读取数据相似:

```
//修改序列数据  
void seq::changeNum(int index, double afterNum) {  
    this->a[index - this->begin] = afterNum;  
}
```

测试数据: A={1, 2, 3} , n=0:2。修改 n=1 时的数据为 9:

```
修改前: begin:0  
end:2  
1 2 3  
修改后: begin:0  
end:2  
1 9 3
```

d) 无限长序列的输入:

实现方法:

在 while(1) 中输入数据后, 通过 vector 的 push_back 函数添加到原 vector 容器中, 实现序列的增长, 每次输入一个值, 输入后进行处理。以加法为例, 输入 1 代表希望进行加法操作:

```
vector<double> u = { 1,2,3 };
seq A(0, 2, u);
//vector<int> u2 = { 3,4,1 };
vector<double> u2 = { 1,2,3 };
seq B(0, 2, u2);

cout << "请选择将要进行的操作: " << endl;
cout << "1.加法 2:乘法 3.相关性计算 " << endl;
int op;
cin >> op;
int in1, in2;
switch (op)
{
case 1:
    cout << "初始化加法结果: ";
    A + B;
    cout << endl;
    break;
```

```
while (1) {
    cin >> in1;
    cin >> in2;
    A.a.push_back(in1);
    B.a.push_back(in2);

    switch (op)
    {
    case 1:
        A + B;
        break;
```

注意:

在本次实验汇总通过最先输入一个 flag 值来确定接下来处理的对象为有限长序列/无限长序列 (因为处理有限/无限长序列步骤不同: 有限长是先通过 init 函数初始化 A, B 序列, 初始化后选择要对序列进行的操作; 但是对无限长序列的操作需要先确定需要进行的操作, 比如进行加法操作, 再不断继续输入数据以使输入后可以直接进行处理然后继续输入)