HM1

学号：

姓名：

问题：

1、输入一个整数，作为新开辟数组的长度，创建该数组，并用随机数填充前一半，后一半为空；

2、输入一个整数，作为插入位置，在该位置插入0；

3、输入一个整数，删除该位置的数据；

4、对数据进行转置。

cpp文件：

#include<iostream>

#include<vector>

#include<cstdlib>

#include<ctime>

using namespace std;

//创建一个vector容器的实例myVector：

typedef vector<int> myVector;

//展示数组的函数：

void display(myVector arr, myVector::iterator itr)

{

for (itr = arr.begin(); itr != arr.end(); itr++)

{

cout << \*itr << '\t';

}

cout << endl << endl;

}

//主函数：

int main() {

srand(time(0));

//输入长度length：

int length;

cout << "Enter an integer as the length of an array.\n";

cin >> length;

//创建数组和迭代器：

myVector arr(length);

myVector::iterator itr = arr.begin();

//向数组前一半填充随机数，后一半填充0

//若长度为奇数，中间位置填0：

for (int i = 0; i < length; i++)

{

if (i < length / 2) //前一半

{

\*itr = rand() % 100; //随机数控制在0-99范围

}

else

{

\*itr = 0;

}

itr++; //迭代器向后移动

}

//展示数组：

display(arr, itr);

//输入插入位置：

int position = 0;

while (position < 1 || position > length) //插入位置在1-length

{

cout << "Enter an integer as the position to insert 0 (1 - " << length << ")\n";

cin >> position;

}

itr = arr.end() - 1;

//从后向前依次将插入位置后的元素向后移动一位

//数组长度不变，最后一个元素在插入后超出范围直接舍弃：

for (int i = 0; i < length - position; i++)

{

\*itr = \*(itr - 1); //当前位置元素变为前一位的元素

itr--; //迭代器向前移动一位

}

\*itr = 0; //此时迭代器已经移动到需要插入的位置，插入0

//展示数组：

display(arr, itr);

//输入删除位置：

position = 0;

while (position < 1 || position > length) //插入位置在1-length

{

cout << "Enter an integer as the position to remove the element (1 - " << length << ")\n";

cin >> position;

}

//删除位置后的元素依次向前移动一位：

for (itr = arr.begin() + position - 1; itr != arr.end() - 1; itr++)

{

\*itr = \*(itr + 1);

}

\*itr = 0; //最后一位补0

//展示数组：

display(arr, itr);

//新建等长的数组：

myVector newArr(length);

myVector::iterator newItr;

itr = arr.end() - 1;

//倒序填入原数组的元素：

for (newItr = newArr.begin(); newItr != newArr.end(); newItr++)

{

\*newItr = \*itr;

if (itr > arr.begin()) //避免迭代器超出范围

{

itr--;

}

}

//展示数组：

display(newArr, newItr);

system("pause");

return 0;

}

截图：

