P46 4.1-5 最大子数组非递归线性运算

```
#include <iostream>
#include <limits>
using namespace std;
template<class T>
T FindMaximumSubarray(T array[], int low, int high, int *leftSubscript, int *rightSubscript){
  bool judgment = true;
  T maxSum = INT_MIN;
  T sum = 0;
  for (int i = low; i <= high; i++){
    if (array[i] >= 0){
      judgment = false;
    }
    if (array[i] > maxSum){
      maxSum = array[i];
      *leftSubscript = i;
      *rightSubscript = i;
    }
  }
  if (judgment = true){
    return maxSum;
  }
  int i = 0;
  for (int j = low; j <= high; i++){
    sum = sum + array[j];
    if (array[j] >= 0){
      if (sum > maxSum){
        maxSum = sum;
        *leftSubscript = i;
        *rightSubscript = j;
      }
    }
    else{
      i = j;
      sum = 0;
    }
  }
}
```

```
int main(){
  int n;
  cout << "输入数组长度：" << endl;
  cin >> n;
  int *arrays = new int[n];
  cout << "输入数组的各个数字：" << endl;
  int j = 0;
  while (j < n){
    cin >> arrays[j];
    j++;
  }
  int leftSubscript;
  int rightSubscript;
  int sum = 0;
  sum = FindMaximumSubarray(arrays, 0, n - 1, &leftSubscript, &rightSubscript);
  cout << leftSubscript << " " << rightSubscript << " " << sum << endl;
  delete arrays;
  return 0;
}
```