**7.10**

#include <iostream>

using namespace std;

constint saleArrSize = 9;

int main(){

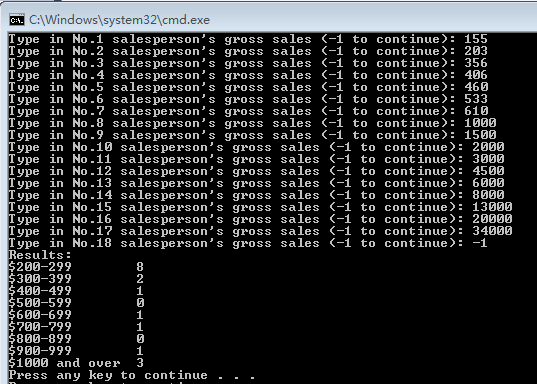
int saleArr[saleArrSize] = {};//存储属于不同收入区间人数的数组

int counter = 1;

while (cin) {

int gross;

cout<<"Type in No."<< counter <<" salesperson's gross sales (-1 to continue): ";

 cin>> gross;

//输入检查

if (gross == -1)

break;

if (gross < 0) {

cout<<"Illegal Input!"<<endl;

continue;

}

int bonus = gross \* 9 / 100;//奖金

int salaryClass = bonus / 100;//确定收入档次

if (salaryClass>= 8)

saleArr[8]++;//总收入大于等于1000

else

saleArr[salaryClass]++;

counter++;

}

cout<<"Results:"<<endl;

for (int i = 0; i< 8; i++)//输出数据

cout<<'$'<< i+2 <<"00-"<< i+2 <<"99\t"<<saleArr[i] <<endl;

cout<<"$1000 and over\t"<<saleArr[8] <<endl;

system("pause");

}

7.11

#include <iostream>

using namespace std;

void bubbleSort(int[], int);

const int arrSize = 10;

int main() {

int arr[arrSize] = {3,5,6,4,8,7,2,1,9,0};

cout<<"Origin: ";

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

cout<<arr[i] <<' ';

cout<<endl;

bubbleSort(arr, arrSize);

cout<<"Sorted: ";

for (inti = 0; i<arrSize; i++)

cout<<arr[i] <<' ';

cout<<endl;

system("pause");

}

void bubbleSort(int arr[], int size) {

for (inti = 0; i< size - 1; i++) {

for (int j = 0; j < size - 1 - i; j++) {

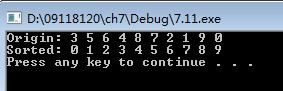
if (arr[j] >arr[j+1])

swap(arr[j], arr[j+1]);

}

}

}

7.12

#include <iostream>

using namespace std;

void bubbleSort(int[], int);

const int arrSize = 10;

int main() {

int arr[arrSize] = {4,8,7,2,0,5,3,1,9,6};

cout<<"Origin: ";

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

cout<<arr[i] <<' ';

cout<<endl;

bubbleSort(arr, arrSize);

cout<<"Sorted: ";

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

cout<<arr[i] <<' ';

cout<<endl;

system("pause");

}

void bubbleSort(int arr[], int size) {

int swapCounter = 0;

for (int i = 0; i< size - 1; i++) {

swapCounter = 0;

for (int j = 0; j < size - 1 - i; j++) {

if (arr[j] >arr[j+1]) {

swap(arr[j], arr[j+1]);

swapCounter++;

}

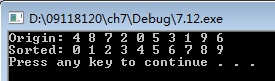
}

if (!swapCounter)//若未交换过则说明已排序完

break;

}

}



7.15

#include <iostream>

using namespace std;

bool find(int, int[], int , int);//查找数组中是否含有某数字

const int arrSize = 20;

int main() {

int arr[arrSize] = {};

int counter = 0;//输入数

int index = 0;//合法元素末索引

while (counter < 20) {

int temp = 0;

cout<<"Type in a 10-100 number(inclusive) ("<< counter + 1 <<" of 20) : ";

cin>> temp;

//输入检查

if (temp < 10 || temp > 100) {

cout<<"Illegal Input!"<<endl;

continue;

} else {

if (counter == 0) {//首项录入

arr[index] = temp;

index++;

} else {//后续项录入

if (!find(temp, arr, 0, counter - 1)) {

arr[index] = temp;

index++;

}

}

}

counter++;

}

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

cout<<arr[i] <<' ';

system("pause");

}

bool find(int finding, int arr[], int start, int end) {

if (arr[start] == finding)

return true;

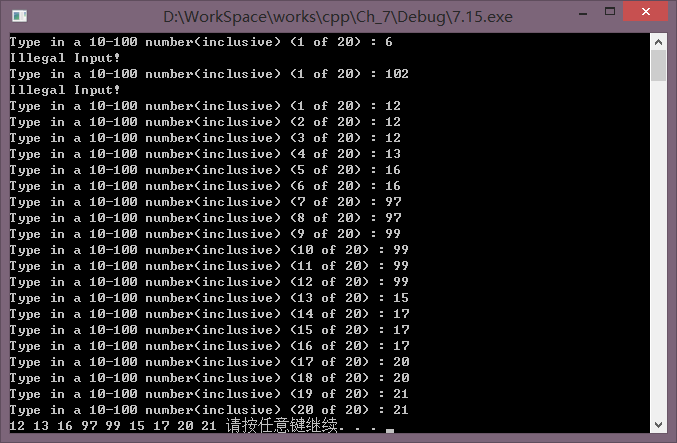
if (start == end)

return false;

else

return find(finding, arr, start + 1, end);

}



7.17

#include<iostream>

#include<ctime>

#include<iomanip>

using namespace std;

int main(){

srand(time(0));

int dice1, dice2;

int sum;

int countArr[11] = {};

for (inti = 0; i< 36000; i++) {

dice1 =1 + rand() % 6;

dice2 =1 + rand() % 6;

sum = dice1 + dice2;

countArr[sum - 2]++;//骰子点数和2-12对应索引0-10

}

cout<<"Sum\tCount\tPercentage"<<endl;

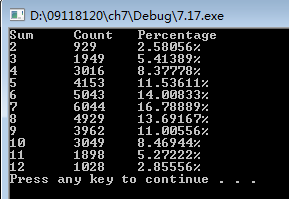
for (int i = 0; i< 11; i++) {

cout<< i+2 <<'\t'<<countArr[i] <<'\t'<< fixed <<setprecision(5) <<countArr[i]/36000.0\*100 <<"%"<<endl;

}

system("pause");

}



7.20

#include <iostream>

#include <string>

using namespace std;

void printPass(int, int); //打印登机牌

int arrangeFirst(bool[]); //安排头等舱座位

int arrangeEconomy(bool[]); //安排经济舱座位

int main() {

bool list[10] = {};//乘客选择及座位数组

cout << "Please type 1 for First Class or type 2 for Economy." << endl;

int i = 1;//第i位乘客

while (i <= 10) {

cout << "The choice of guest No." << i << ": ";

int temp;

cin >> temp;

int seat;

switch (temp) {

case 1:

seat = arrangeFirst(list);

break;

case 2:

seat = arrangeEconomy(list);

break;

default:

cout << "Illegal input!" << endl;

seat = -1;

break;

}

if (seat == -1)continue;

printPass(i, seat + 1);

i++;

}

system("pause");

}

void printPass(int number, int seat) {

cout << "Guest No." << number << ", Your boarding pass is:\n";

cout << "Class: " << (seat <= 5 ? "First" : "Economy") << endl;

cout << "Seat: " << seat << endl;

}

int arrangeFirst(bool list[]) {

int seat = 0;//头等舱第一个座位索引为0

while (list[seat] && seat <= 4)seat++;//寻找空座位

if (seat >= 5) {//满员

char answer;

cout << "The First Class is full. Would you please switch to Economy? (y/n)" << endl;

cin >> answer;

switch (answer) {

case 'y':

return arrangeEconomy(list);

case 'n': default:

cout << "Next flight leaves in 3 hours." << endl;

return -1;

}

} else {

list[seat] = true;

return seat;

}

}

int arrangeEconomy(bool list[]) {

int seat = 5;//经济舱第一个座位索引为5

while (list[seat] && seat <= 9)seat++;//寻找空座位

if (seat >= 10) {//满员

char answer;

cout << "The Economy is full. Would you please switch to First Class? (y/n)" << endl;

cin >> answer;

switch (answer) {

case 'y':

return arrangeFirst(list);

case 'n': default:

cout << "Next flight leaves in 3 hours." << endl;

return -1;

}

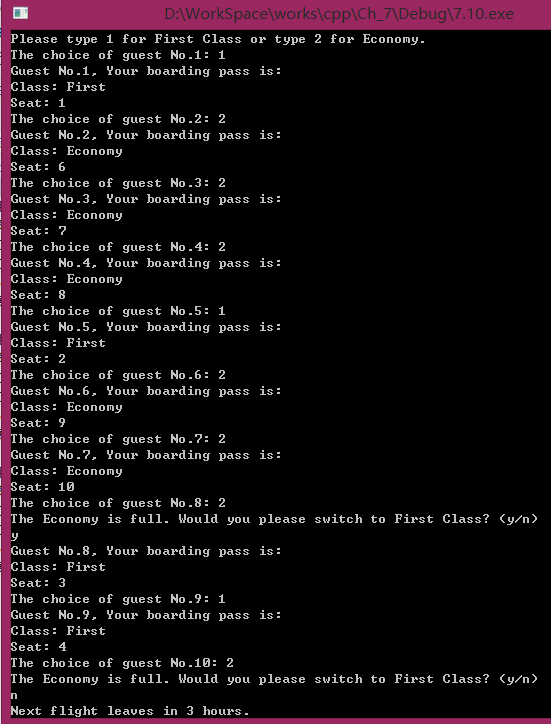
} else {

list[seat] = true;

return seat;

}

}



7.21

// Ex. 7.21: Ex07\_21.cpp

// What does this program do?

#include <iostream>

using namespace std;

void someFunction(int[], int, int);

int main()

{

const int arraySize = 10;

int a[arraySize] = { 1,2,3,4,5,6,7,8,9,10 };

cout << "The values in the array are:" << endl;

someFunction(a, 0, arraySize);

cout << endl;

}

//What does this function do?

void someFunction(int b[], int current, int size)

{

if (current < size)

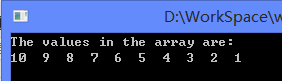
{

someFunction(b, current + 1, size);

cout << b[current] << " ";

}

}



\*该函数反向输出这个数组

7.32

#include<iostream>

#include<string>

using namespace std;

bool testPalindrome(string, int, int);

int main() {

string str;

while (cin) {

getline(cin, str);

cout<< (testPalindrome(str, 0, str.length() - 1) ? "Palindrome!" :"Not a palindrome!") <<endl;

}

}

bool testPalindrome(string str, int startIndex, int endIndex) {

//每次检查首尾2个字符是否一致

int temp1 = startIndex, temp2 = endIndex;

//忽略非字母

while (!((str[temp1]>= 'a'&&str[temp1]<= 'z') || (str[temp1]>= 'A'&&str[temp1]<= 'Z')))

temp1++;

while (!((str[temp2]>= 'a'&&str[temp2]<= 'z') || (str[temp2]>= 'A'&&str[temp2]<= 'Z')))

temp2--;

if (temp1 > temp2)//若光标移动过数组中间则停止

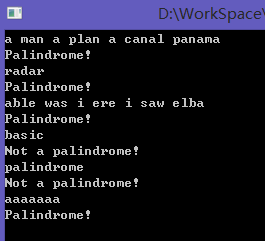
return true;

if (str[temp1] != str[temp2])//判断首尾是否相同

return false;

else

return testPalindrome(str, temp1 + 1, temp2 - 1);//继续检查下一组首尾字符

}

7.36

#include<iostream>

#include<string>

using namespace std;

void stringReverse(char[], int);

int main() {

while (cin) {

char str[1024] = {};

cin.getline(str, 1024);

stringReverse(str, 0);

cout<<endl;

}

}

void stringReverse(char str[], int start) {

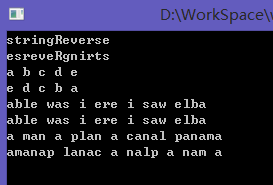
if (str[start] == '\0')

return;

stringReverse(str, start + 1);

cout<<str[start];

}



7.37

#include<iostream>

using namespace std;

int\* recursiveMinimum(int[], int, int);

const int arrSize = 10;

int main() {

int arr[arrSize] = { 16,20,8,45,63,12,89,48,76,32 };

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

cout<<arr[i] <<' ';

cout<<endl;

cout<<"The minimum is "<< \*recursiveMinimum(arr, 0, arrSize - 1) <<endl;

system("pause");

}

int\* recursiveMinimum(int arr[], int start, int end) {

if (start == end)

return &arr[start];

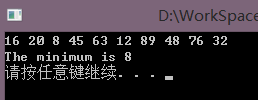
if (arr[start] >= arr[end])

return recursiveMinimum(arr, start + 1, end);

else

return recursiveMinimum(arr, start, end - 1);

}



7.40

#include<iostream>

#include<vector>

using namespace std;

int recursiveMinimum(const vector<int>&, int, int);

const int arrSize = 10;

int main() {

vector<int> arr;

cout<<"Type in 10 numbers:";

for (int i = 0; i<arrSize; i++) {

int temp;

cin>> temp;

arr.push\_back(temp);

}

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

cout<< arr.at(i) <<' ';

cout<<endl;

cout<<"The minimum is "<<recursiveMinimum(arr, 0, arrSize - 1) <<endl;

system("pause");

}

int recursiveMinimum(const vector<int>&arr, int start, int end) {

if (start == end)

return arr.at(start);

if (arr.at(start) >= arr.at(end))

return recursiveMinimum(arr, start + 1, end);

else

return recursiveMinimum(arr, start, end - 1);

}

