(1)

Header file:

template <typename T>

int find(T val, T \*list, int size) {

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

if (list[i] == val)

return i;

}

return -1;

}

Source-code file:

#include<iostream>

#include"test01.h"

using namespace std;

int main() {

int val, size;

int arrays[10] = { 0,1,2,3,4,5,6,7,8,9 };

size = 10;

val = 2;

cout<<find(val,arrays,size);

return 0;

}

(2)

Header file:

#pragma once

template <class T>

class employes {

T scores, days;

public:

employes(T score, T day) {

scores = score;

days = day;

}

T sumscore() {

return scores\*days;

}

};

Source-code file:

#include<iostream>

#include"test02.h"

using namespace std;

int main() {

employes<int> a1(1,4) ;//create a object with int type

cout << a1.sumscore() << endl;;

employes<double> a2(2.3, 6);

cout << a2.sumscore();

system("pause");

return 0;

}