计算机科学系　　2012 第一学期

《　程序设计 I　》期末考试试题答案(A)

任课教师：吴维刚　刘聪　刘晓铭　考试形式：闭卷　考试时间：2 小时

1. **CABDB　DCBDA　DBDCA　CBCDA**

2. **每改对 1 个错误 2 分，10 个满分。**

   line 4: void -> int
   line 8: const int N = 10;
   line 11: j = 0;
   line 11: j<N;
   line 13: avg + = data[j]/10 或者 avg + = (data[j]-avg)/(j+1);
   line 16: index = 0;
   line 19: i < N

   line 1: const char array[], length should be a parameter
   line 4: array[0] → array[i]
   line 5: array[i] is a constant
   line 12: array index out of bound
   line 13: first argument: cannot convert int * to char * (line 12, int → char)

3.

(1)

```
1 1
1 2
2 1
2 4
```

(2)

```
1
abcabcabc
4
abcabcabc
7
abcabcabc
```

4.

```
number == 1
number % factor != 0
getFactors(number / factor, false)
array[i], true
a, 5
```

5.

Function matching is a task for the compiler to determine the function to be called in each function call express. It is determined according to the function name and the types of its parameters in the function call

expression.

If there is no exact match for a function, the compiler will look for match after conversions of arguments. In the case of no exact match and there are multiple matches after conversion of argument, the compiler cannot determine which function to be called in the function call expression. This is when the compiler reports a *ambiguous function* call error.

```
int add(int a, int b) {
    return a + b;
}

double add(double a, double b) {
    return a + b;
}

int main() {
    // call of overloaded 'add(int, double)'
    // is ambiguous
    int b = add(1, 2.0);
}
```

The commonalities between array and pointer:

they are both address,

and the operator +, *, and [] are applicable to them.

The difference is that:

array is a constant address and pointer is a address variable,

and, therefore, the left operand of += and only be a pointer.

```
int array[] = {1, 2, 3};
int * pointer = array;
// + is applicable to both
cout << *(array + 1) << endl;
cout << *(pointer + 1) << endl;
// * is applicable to both
cout << *array << endl;
cout << *pointer << endl;
// [] is applicable to both
cout << array[1] << endl;
cout << *pointer[1] << endl;

// However, the left operand of
// += can only be a pointer
pointer += 1;
```

6.

```
int length(char string[]) {
    for (int i = 0; ; ++ i) {
        if (string[i] == 0) return i;
    }
}

void remove(char string[], int index) {
    int len = length(string);
    for (int i = index; i < len; ++ i) {
        string[i] = string[i + 1];
    }
}

bool isDigit(char c) {
    return (c >= '0' && c <= '9');
}

void removeDigits(char string[]) {
    for (int i = length(string) - 1; i >= 0 ; -- i) {
```

```
        if (isDigit(string[i])) remove(string, i);
    }
}
```

```
#include <iostream>
using namespace std;

bool isSolution1(int x) {
    return (x * x - 20 * x + 75 == 0);
}

bool isSolution2(int x) {
    return (x % 5 == 0);
}

int main() {
    int n;
    cin>>n;
    for (int x = 1; x < n; ++ x) {
        if (isSolution1(x) && isSolution2(x)) {
            cout << "x = " << x << endl;
        }
    }
}
```