计 算 机 科 学 系 　　　2011 第 一 学 期

《 程 序 设 计 **I**》 期 末 考 试 试 题 (**A**)

任 课 教 师 ： 吴 维 刚　 刘 聪　 林 瀚 　　　　考 试 形 式 ： 闭 卷　 考 试 时 间 ： **2** 小 时

年级： **11**　 班别：**1、2、3** 专业： 计算机科学与技术　　 姓名：_____ 学号：_____成绩

1. Single Answer Questions (only one choice is correct). 20points.

Choose the alternative(s) that best completes the statement or answers the question.

1) Which of the following expressions will NOT yield 0.5?
   A) 1.0 / 2   B) (double)1 / 2   C) 1 / (double)2   D) (double)(1 / 2)

2) Which of the following assignment statements will cause compile error?
   A) char c = 'A';   B) char c = 99;   C) char c = "a";   D) char c = 'A' + 'B';

3) Analyze the following program fragment:

   int n = 10000 * 10000 * 10000;
   cout << "n is " << n << endl;

   A) The program displays n is 1000000000000
   B) This causes an overflow and the program is aborted.
   C) This causes an underflow and the program is aborted.
   D) This causes an overflow and the program continues to execute because C++ does not report errors on overflow.

4) Which of the following Boolean expressions is correct?
   A) true || x > 5   B) x <=3 && x<>1   C) 5<x<10   D) x=<1 || 2==x

5) To check whether a char variable ch is an uppercase letter, you write _____.
   A) ch >= 'A' && ch <= 'Z'   B) ch >= 'A' && ch >= 'Z'
   C) ch >= 'A' || ch <= 'Z'   D) 'A' <= ch <= 'Z'

6) Suppose x = 1. What is x after evaluating the expression (y > 0) && (x++ > 0)?
   A) 0   B) 1   C) 2   D) It depends on what y is.

7) What is y after the following switch statement is executed?

```
x=1;
switch(x+=1){
    case 1: y =0;
    case 2: y = 1;
    default: y+=1;
}
```
A) 0   B) 1   C) 2   D) 3

8) Which of the following operators are right-associative?

A) =   B) %   C) /   D) &&

9) What is the number of iterations in the following loop:

```
for (int i = 1; i < 10; i++)
{
   // iteration
}
```
A) 8   B) 9   C) 10   D) 11

10) Analyze the following program fragment:

```
double sum = 0;
double d = 0;
while (d != 10.0)
{
  d += 0.1;
  sum += d;
}
```
A) The program does not compile because sum and d are declared double, but assigned with integer value 0.
B) The program never stops because d is always 0.1 inside the loop.
C) The program may not stop because of the phenomenon referred to as numerical inaccuracy for operating with floating-point numbers.
D) After the loop, sum is 0 + 0.1 + 0.2 + 0.3 + ... + 1.9

11) Each time a function is invoked, the system stores parameters and local variables in an area of memory, known as _____, which stores elements in last-in-first-out fashion.

A) a heap   B) a stack   C) static area   D) an array

12) If a parameter is a reference variable, this parameter becomes an alias for the original variable. This is referred to as _____.

A) function invocation   B) pass by value   C) pass by reference   D) pass by name

13) What is k AFTER the following block executes?

2

```
  {
   int k = 2;
  }
```
A) 0   B) 1   C) 2   D) k is not defined outside the block.

14) Suppose i = 5, which of the following can NOT be used as an index for array double t[100] ?

A) i   B) i + 4.5   C) i + 94   D) rand()%100

15) Analyze the following code.

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

int main()
{
  int x[3];
  cout << "x[0] is " << x[0];

  return 0;
}
```

A) The program has a compile error because the size of the array wasn't specified when declaring the array.

B) The program has a runtime error because the array elements are not initialized.

C) The program runs fine and displays x[0] is 0.

D) x[0] has an arbitrary value.

16) Given the following two arrays:
   char s1[] = {'a', 'b', 'c'};
   char s2[] = "abc";
   Which of the following statements is correct?

A) Both s1 and s2 have size 3.

B) s1 has size 3, while s2 has size 4.

C) s1 has size 4, while s2 has size 3.

D) Both s1 and s2 have size 4.

17) When you pass an array to a function, the function receives _____.

A) a copy of the array

B) a copy of the first element

C) the reference/address of the array

D) the length of the array

18) Which of the following function declaration is correct?

A) int f(int[][] a, int rowSize, int columnSize);

B) int f(int a[][], int rowSize, int columnSize);

C) int f(int a[][3], int rowSize);

D) int f(int a[3][], int rowSize);

19)   Suppose you declare int list[10]; which of the following is NOT true?

A) list is the address of list[0].

B) &list[0] is the address of list[0].

C) *(list + 4) is list[4].

D) int(&list[1]) – int(&list[0]) is 1.

20)   Analyze the following recursive function:

```
int factorial(int n)
{
    return n * factorial(n - 1);
}
```

A) Invoking factorial(0) returns 0.

B) Invoking factorial(1) returns 1.

C) The function runs infinitely and causes some memory problem.

D) The function runs infinitely, but does not cause any memory problem.

2. Mistakes identification. 20points.

The following C++ code will not compile/execute correctly. Please find out and correct the mistakes (will cause compiling or running errors).

(1) The following program summarizes integers (between 1 and 100) from input. (10')

	#include <iostream>
	using namespace std;
1)	int main(){
2)	int value;
3)	int sum;
4)	while(true){
5)	cout << "Enter an integer [1-100], -1 to exit:\n";
6)	cin>>value;
7)	if (value = -1)
8)	break;
9)	else if(1<=value<=100)
10)	sum += value;
11)	else
12)	cout<<"A wrong number.\n";
13)	}
14)	cout<< int sum<<endl;
15)	}

(2) The program reverses a given array by exchanging the elements from left to right. (10')

	#include <iostream>
	using namespace std;
1)	void reverse(int list[], int size){
2)	for (int i=0,j=size-1; i<size/2; i++,j--){
3)	int temp = list[i];
4)	list[i] = list[j];
5)	list[j] = list[i];
6)	}
7)	return 0;
8)	}
9)	int main(){
10)	int SIZE = 8;
11)	int myList[SIZE];
12)	for(int i = 1; i<=SIZE; i++)
13)	*(myList++) = i;
14)	reverse(myList[], SIZE);
15)	for (int i = 0; i <= SIZE; i++)
16)	cout << myList[i] << " ";
17)	return 0;
18)	}

3. Program output analysis. 20points.

Write down the output for the following programs.

(1)	(2)
| ```cpp
#include <iostream>
using namespace std;

int main() {
    int count = 0;
    while (count < 100) {
        count += 2;
        count *= 3;
        cout << count << endl;
    }
}
``` | ```cpp
#include <iostream>
using namespace std;

int product(int value) {
    if (value <= 3) return value;
    int value1 = value / 2;
    int value2 = value - value1;
    return product(value1)*
            product(value2);
}

int main() {
    cout << product(3) << endl;
    cout << product(5) << endl;
    cout << product(10) << endl;
}
``` |

```
(3)
#include <iostream>
using namespace std;

int sum(int array[], int length){
    int sum = 0;
    for (int i = 0; i < length; ++ i){
        sum += array[i];
    }
    return sum;
}
```

```
int max(int array[], int length){
    int max = array[0];
    for (int i = 1; i < length; ++ i){
        if (max < array[i])
            max = array[i];
    }
    return max;
}
int main() {
    int array[5] = { 1, 22, 333, 44, 5 };
    cout << sum(array, 5) << endl;
    cout << max(array, 5) << endl;
    int result = sum(array, 5) –
        max(array, 5);
    cout << result<<endl;
}
```

```
(4)
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

void append(char array[], const char c) {
    int length = strlen(array);
    array[length] = c;
    array[length + 1] = 0;
}
bool contains(char array[], char c) {
    int length = strlen(array);
    for (int i = 0; i < length; ++ i) {
        if (c == array[i]) return true;
    }
    return false;
}


void toIdentifierName(char array[],
                            char name[]){
  name[0] = 0;
  bool isFirst = false;
  int length = strlen(array);
  char spaces[] = " \t\r\n";
```

```
  for (int i = 0; i < length; ++ i){
    if (contains(spaces, array[i]))
      isFirst = true;
    else {
      if (isFirst) {
        append(name, toupper(array[i]));
        isFirst = false;
      }else
        append(name, tolower(array[i]));
    }
  }
}

int main(){
    char array1[] = "This";
    char array2[] = "What is up";
    char array3[] = "this is a string";
    char name[100];
    toIdentifierName (array1, name);
    cout << name << endl;
    toIdentifierName (array2, name);
    cout << name << endl;
    toIdentifierName (array3, name);
    cout << name << endl;
}
```

4. Fill-in-the-blank. 20points.

Please fill in the blanks in the following programs to complete them.

(1)This program finds the maximum number in an array.

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

int   ①  (  ②  , int length) {
    if (length == 1)   ③  ;

    int max =   ④  ;
    int last = array[length - 1];
    return (max > last ? max : last);
}

int main() {
    int array[] = { 1,2,4,8,6,4,2,1 };
    int max = findMax(array,   ⑤  );
    cout << "max = " << max;
}
```

(2) This program finds the maximum integer value represented by 3 consecutive characters in a string. For example, if the input is "92319373", the output is 937.

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

int maxSubstring(char array[], int n) {
    int loops =   ①  ;
    int max = 0;
    for (int i = 0; i < loops; ++ i) {
        char * p = new char[n + 1];
        strncpy(  ②  );

          ③  ;
        int value = atoi(p);

          ④  ;
        if (max < value) max = value;
    }
      ⑤  ;
}

int main() {
    char string1[] =
        "47923193773 2622477666";
    cout << maxSubstring(string1, 3);
}
```

5. Programming design. 20points.

(1) Split a string

Input two strings. Please find out all the substrings in the first string that are separated by the characters in the second string.

Print these substrings each in a new line in the order they appears in the first string.

Assume all strings are shorter than 100 characters.

输入两个字符串。

用第二个字符串中的字符作为分隔符，分隔第一个字符串，打印出分隔后的所有子字符串，每个子串一行。空串不需要输出。

程序中可以假设所有字符串的长度都少于100个字符。

Example input:
09A105u3FB9A53R26
AFSBT2

Example output:
09
105u3
9
53R
6

(2) Retain each character once.

Input a string and remove all characters that have appeared in the same string.

Note that the first appearance of each character should not be removed.

Print all the remaining characters in a line and in the same order they first appear in the string.

Assume the input string is shorter than 100 characters.

输入一个字符串。删除所有已经出现过的字符。

注意当一个字符第一次出现的时候不应被删除。

程序中可以假设字符串的长度都少于100个字符。

Example input:
09A1053FB953126D2DBC45C9B22DC2

Example output:
09A153FB26DC4