一、不定项选择题  
1、以下程序的打印结果是（）

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. #include<iostream>
2. using namespace std;
4. void swap\_int(**int** a , **int** b)
5. {
6. **int** temp = a;
7. a = b;
8. b = temp;
9. }
11. void swap\_str(**char**\* a , **char**\* b)
12. {
13. **char**\* temp = a;
14. a = b;
15. b = temp;
16. }
18. **int** main(void)
19. {
20. **int** a = 10;
21. **int** b = 5;
22. **char**\* str\_a = "hello world";
23. **char**\* str\_b = "world hello";
24. swap\_int(a , b);
25. swap\_str(str\_a , str\_b);
26. printf("%d %d %s %s\n", a , b , str\_a , str\_b);
28. return 0;
29. }

#include<iostream>

using namespace std;

void swap\_int(int a , int b)

{

int temp = a;

a = b;

b = temp;

}

void swap\_str(char\* a , char\* b)

{

char\* temp = a;

a = b;

b = temp;

}

int main(void)

{

int a = 10;

int b = 5;

char\* str\_a = "hello world";

char\* str\_b = "world hello";

swap\_int(a , b);

swap\_str(str\_a , str\_b);

printf("%d %d %s %s\n", a , b , str\_a , str\_b);

return 0;

}

A、10 5 hello world world hello     B、10 5 world hello hello world  
C、5 10 hello world world hello     D、5 10 hello world world hello  
2、以下程序打印的两个字符分别是（A）

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. typedef struct object object;
2. struct object
3. {
4. **char** data[3];
5. };
7. **int** main(void)
8. {
9. object obj\_array[3] = { {'a','b','c'},
10. {'d','e','f'},
11. {'g','h','i'} };
12. object\* cur =  obj\_array;
13. printf("%c %c\n", \*(**char**\*)((**char** \*)(cur)+2) , \*(**char**\*)(cur+2));
15. return 0;
16. }

typedef struct object object;

struct object

{

char data[3];

};

int main(void)

{

object obj\_array[3] = { {'a','b','c'},

{'d','e','f'},

{'g','h','i'} };

object\* cur = obj\_array;

printf("%c %c\n", \*(char\*)((char \*)(cur)+2) , \*(char\*)(cur+2));

return 0;

}

A、c g           B、b d          C、g g          D、g c  
3、C/C++语言：请问在64位平台机器下 sizeof(string\_a) , sizeof(string\_b)大小分别是（A）

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. **char** \*string\_a = (**char** \*)malloc(100\*sizeof(**char**));
2. **char** string\_b[100];

char \*string\_a = (char \*)malloc(100\*sizeof(char));

char string\_b[100];

A、8 100      B、100 8      C、100 100      D、8 8  
4、假设二叉排序树的定义是：1、若它的左子树不为空，则左子树所有节点均小于它的根节点的值；2、若右子树不为空，则右子树所有节点的值均大于根节点的值；3、它的左右子树也分别为二叉排序树。下列哪种遍历之后得到一个递增有序数列（B）  
A、前序遍历    B、中序遍历     C、后序遍历      D、广度遍历  
5、往一个栈顺序push下列元素：ABCDE，其pop可能的顺序，下列不正确的是（C）  
A、BACDE    B、ACDBE     C、AEBCD   D、AEDCB  
6、1100|1010 ， 1001^1001 ， 1001&1100分别为（A）  
A、1110  0000   1000   B、1000  1001  1000  
C、1110  1001   0101   D、1000  1001  1000  
7、二叉树是一种树形结构，每个节点至多有两颗子树，下列一定是二叉树的是（AC）  
A、红黑树     B、B树      C、AVL树      D、B+树  
8、int A[2][3] = {1,2,3,4,5,6}; ， 则A[1][0]和\*(\*(A+1)+1)的值分别是（A）  
A、4 5     B、4 3        C、3 5       D、3 4  
9、序列16 14 10 8 7 9 3 2 4 1的说法下面哪一个正确（A）  
A、大顶堆   B、小顶堆  C、不是堆   D、二叉排序树   
10、输入若已经是排好序的，下列排序算法最快的是（A）  
A、插入排序   B、Shell排序   C、合并排序   D、快速排序  
11、一种既有利于短作业又兼顾长期作业的调度方式是（D）  
A、先来先服务     B、均衡调度      C、最短作业优先      D、最高响应比优先  
12、同一进程下的线程可以共享（B）  
A、stack        B、data section        C、register set      D、thread ID  
13、系统中的“颠簸”是由（B）引起的。  
A、内存容量不足      B、缺页率高      C、交换信息量大       D、缺页率反馈模型不正确  
14、8瓶酒一瓶有毒，用人测试。每次测试结果8小时后才会得出，而你只有8个小时的时间。问最少需要（B）人测试？  
A、2         B、3         C、4            D、6  
是３个人，如果你学过数的２进制编码，就容易说了：  
８瓶酒的编码如下：  
0: 000  
1: 001  
2: 010  
3: 011  
4: 100  
5: 101  
6: 110  
7: 111  
３个人分别喝３个位上为１的编码，所以：  
第一个：1,3,5,7  
第二个：2,3,6,7  
第三个：4,5,6,7  
把中毒的人的位填１的二进制数，就是毒酒的编号。  
15、下列关于网络编程错误的是（AB）  
A、TCP建立和关闭连接都只需要三次握手  
B、UDP是可靠服务  
C、主动关闭的一端会出现TIME\_WAIT状态  
D、服务端编程会调用listen()，客户端也可以调用bind()  
16、进程间通讯有哪几种形式（ABCD）  
A、Socket  
B、Pipe  
C、Shared memory  
D、Signal  
17、TCP/UDP下面正确的是（AC）  
A、TCP provide connection-oriented,byte-stream service;  
B、Both TCP and UDP provide reliability service;  
C、TCP also provides flow control;  
D、Both TCP and UDP provide retransmission mechanism;  
18、分布式系统设计包括（ABCDE）  
A、容错，design for fault  
B、多数据中心的数据一致性  
C、数据/服务可靠性  
D、可扩展性  
E、要满足ACID特性  
19、10个不同的球，放入3个不同的桶内，共有（C）种方法。  3^10  
A、1000      B、720     C、59049     D、360  
20、87的100次幂除以7的余数是多少（D）  
A、1     B、2     C、3     D、4  
二、简答题：  
1、  
（1）请描述进程和线程的区别？  
（2）多线程程序有什么优点、缺点？  
（2）多进程程序有什么优点、缺点？与多线程相比，有何区别？  
  
2、编程题：  
写代码，反转一个单链表，分别以迭代和递归的形式来实现

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. typedef struct node LinkNode;
2. struct node
3. {
4. **int** data;
5. LinkNode\* next;
6. };

typedef struct node LinkNode;

struct node

{

int data;

LinkNode\* next;

};

// 返回新链表头节点  
LinkNode \*reverse\_link(LinkNode \*head)  
LinkNode \*reverse\_link\_recursive(LinkNode \*head)

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. // 返回新链表头节点
2. LinkNode \*reverse\_link(LinkNode \*head)
3. {
4. if(head == NULL)
5. return NULL;
6. LinkNode \*prev , \*curr , \*reverse\_head , \*temp;
7. prev = NULL , curr = head;
8. while(curr->next)
9. {
10. temp = curr->next;
11. curr->next = prev;
12. prev = curr;
13. curr = temp;
14. }
15. curr->next = prev;
16. reverse\_head = curr;
17. return reverse\_head;
18. }
20. LinkNode \*reverse\_link\_recursive(LinkNode \*head)
21. {
22. if(head == NULL)
23. return NULL;
24. LinkNode \*curr , \*reverse\_head , \*temp;
25. if(head->next == NULL)    // 链表中只有一个节点，逆转后的头指针不变
26. return head;
27. else
28. {
29. curr = head;
30. temp = head->next;    // temp为(a2,...an)的头指针
31. reverse\_head = reverse\_link\_recursive(temp);   // 逆转链表(a2,...an),并返回逆转后的头指针
32. temp->next = curr;    // 将a1链接在a2之后
33. curr->next = NULL;
34. }
35. return reverse\_head;      // (a2,...an)逆转链表的头指针即为(a1,a2,...an)逆转链表的头指针
36. }

// 返回新链表头节点

LinkNode \*reverse\_link(LinkNode \*head)

{

if(head == NULL)

return NULL;

LinkNode \*prev , \*curr , \*reverse\_head , \*temp;

prev = NULL , curr = head;

while(curr->next)

{

temp = curr->next;

curr->next = prev;

prev = curr;

curr = temp;

}

curr->next = prev;

reverse\_head = curr;

return reverse\_head;

}

LinkNode \*reverse\_link\_recursive(LinkNode \*head)

{

if(head == NULL)

return NULL;

LinkNode \*curr , \*reverse\_head , \*temp;

if(head->next == NULL) // 链表中只有一个节点，逆转后的头指针不变

return head;

else

{

curr = head;

temp = head->next; // temp为(a2,...an)的头指针

reverse\_head = reverse\_link\_recursive(temp); // 逆转链表(a2,...an),并返回逆转后的头指针

temp->next = curr; // 将a1链接在a2之后

curr->next = NULL;

}

return reverse\_head; // (a2,...an)逆转链表的头指针即为(a1,a2,...an)逆转链表的头指针

}

3、给一个数组，元素都是整数（有正数也有负数），寻找连续的元素相加之和为最大的序列。  
如：1、-2、3、5、-4、6 连续序列3、5、-4、6的和最大。  
如元素全为负数，则最大的和为0，即一个也没有选。  
/\*  
array[]     输入数组  
n           数组元素个数  
            返回最大序列和  
\*/  
int find\_max\_sum(int array[] , int n)

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/8015964)

1. **int** find\_max\_sum(**int** array[] , **int** n)
2. {
3. **int** i , max , sum;
4. sum = max = array[0];
5. for(i = 1 ; i < n ; ++i)
6. {
7. if(sum < 0)
8. sum = array[i];
9. else
10. sum += array[i];
11. if(sum > max)
12. max = sum;
13. }
14. if(max < 0)
15. max = 0;
16. return max;
17. }

int find\_max\_sum(int array[] , int n)

{

int i , max , sum;

sum = max = array[0];

for(i = 1 ; i < n ; ++i)

{

if(sum < 0)

sum = array[i];

else

sum += array[i];

if(sum > max)

max = sum;

}

if(max < 0)

max = 0;

return max;

}

三、设计题  
1、设计一个图片存储系统：假设有一个相册系统，每个用户不限制上传的图片数目，每张相片压缩后都在1M以内，需求如下：  
（1）文件数量太大，采用传统的文件系统存储导致目录系统非常臃肿，访问速度变得缓慢；  
（2）单机存储容量已经远远不能承载所有的文件；  
（3）上传之后，用户只有读取操作和删除操作，不支持修改，整个系统读写比例10:1  
思路：可以使用分布式的文件系统，觉得hadoop的HDFS很符合要求，这是hadoop对googleGDFS的实现。