

2016~2017 学年第一学期期末考试试卷

《计算机软件技术基础 2》(C++、64 学时) (A 卷 共 7 页)

(考试时间: 2017 年 1 月 11 日)

题号	一	二	三	四	五	成绩	核分人签字
得分							

一、单选题, 将答案填在下面的表格中, 直接标在题上不记分 (每小题 1 分, 共 20 分)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

1. 设 $A[0 \dots m-1]$ 为顺序循环队列, 若 $front$ 和 $rear$ 分别表示队头及队尾, 则当前队列中元素的数目是 ()。
- A. $(rear-front+m) \%m$ B. $rear-front+1$
 C. $rear-front-1$ D. $rear-front$
2. 某二叉树的中序遍历序列为 A,B,C,D,E,F,G, 后序序列为 B,D,C,A,F,G,E, 则其前序序列是 ()。
- A. E,G,F,A,C,D,B B. E,A,C,B,D,G,F
 C. E,A,G,C,F,B,D D. 以上都不对
3. 冒泡排序比较次数最少的情况及最少比较次数为 ()。
- A. 正序排列, 1 次 B. 逆序排列, 1 次
 C. 正序排列, $n-1$ 次 D. 逆序排列, $n-1$ 次
4. 在 C++ 程序中, 表示注释的符号是 ()。
- A. # B. \ \ C. /* * / D. ->
5. 数学关系 $x>8$ 且 $x\leq 12$ 写成 C++ 语言表达式是 ()。
- A. $8 < x \& x \leq 12$ B. $8 < x \leq 12$ C. $x > 8 \parallel x \leq 12$ D. $x > 8 \& x \leq 12$
6. 设有定义: int a, b; 表达式 $(a=3, b=5, a>b)? a++: b++, a+b$ 的值是 ()。
- A. 3 B. 8 C. 10 D. 9
7. 与 $if(a!=0)$ 等价的是 ()。
- A. if(a) B. if(a=0) C. if(!a) D. if(a>=0)
8. 类 A 的成员函数为 void Set(A &a), 其中 A &a 的含义为 ()。
- A. 变量 A 与 a 相互作为 Set() 的形参 B. 将 a 的函数地址赋给 Set
 C. a 是类 A 的对象引用, 用作函数 Set() 的形参 D. 指向类 A 的指针为 a
9. 已知 char a[]="Happy", b[]={'H', 'a', 'p', 'p', 'y'}; 正确的是 ()。
- A. 数组 a 的长度大于数组 b 的长度 B. 数组 a 和数组 b 的内容相同
 C. 数组 a 的长度小于数组 b 的长度 D. 数组 a 和数组 b 的长度相同
10. 用 new 运算符创建一维数组的正确形式是 ()。
- A. int *p=new float[10]; B. int *p=new int[10];
 C. int *p=new a[10]; D. int *p=new int[5]={1,2,3,4,5,6};
11. 已知: int a[]={1, 2, 3, 4, 5, 6}, *p=a; 下面语句执行后输出*p 值为 5 的是 ()。
- A. p+=4; ++p; B. p+=3; *(p++); C. p+=4; *++p; D. p+=5; *p++;
12. 设类 B 是类 A 的派生类, 当创建一个类 B 的对象时, 构造函数调用顺序是 ()。
- A. 先调类 B 的构造函数, 再调类 A 的构造函数 B. 不调用类 A 的构造函数
 C. 先调类 A 的构造函数, 再调类 B 的构造函数 D. 视定义时类 A 和 B 出现位置
13. 下列常量中不正确的是 ()。
- A. -0.35e-50 B. 0x87AB C. 059 D. "abcd"
14. 下列变量名中不正确的是 ()。
- A. x007 B. _tt30 C. www D. a+b
15. 下列函数中能够实现向文件中写数据的是 ()。
- A. getline B. read C. seekp D. put
16. 在函数体中, return 语句 ()。
- A. 必须返回一个值 B. 可以返回多个值
 C. 可以多次出现 D. 只能出现一次
17. 输出语句 $cout << "C++" << endl$; 中的 cout 为 ()。
- A. 函数名 B. 对象名 C. 变量名 D. C++ 关键字

18. 下列函数原型声明中错误的是 ()。

- A. void Fun(int x=0; int y=0);
- B. void Fun(int, int);
- C. void Fun(int x, y=0);
- D. void Fun(int x, int y);

19. 设指针 p 指向单链表中的结点 m, 若要删除 m 之后的一个结点 (假设该结点存在), 则能够保持正确指向关系的操作是 ()。

- A. p->next=p;
- B. p->next=p->next->next;
- C. p=p->next;
- D. p=p->next->next;

20. 程序控制一般分为三种基本结构, 包括分支、循环和 ()。

- A. 跳转
- B. 顺序
- C. 递归
- D. 返回

二、写出以下程序的运行结果 (每小题 4 分, 共 24 分)

1. #include <iostream.h>

程序运行结果:

```
void main()
{
    int a[4], i, j, k;
    for (i=0; i<4; i++) a[i] = 0;
    k = 3;
    for (i=0; i<k; i++)
        for (j=0; j<=k; j++) a[j] = a[i] + 2;
    cout << a[1] << " " << a[3] << endl;
}
```

2. #include <iostream.h>

程序运行结果:

```
long func(int b, int n)
{
    if (n==0 || n==1) return 1;
    else
    {
        cout << b << " " << n << endl;
        return func(b-1, n-2) + func(b-2, n-1);
    }
}

void main()
{
    cout << func(8,4) << endl;
}
```

3. #include <iostream.h>

程序运行结果:

```
int sum(int p[], int n);
void main()
{
    int data[ ][3]={{ {1,2},{3,4,5},{6,7},{8,9} }};
    for(int i=0; i<4; i++)
        cout << sum(data[i],3) << endl;
}

int sum(int p[], int n)
{
    int i;
    static int s=0;
    for(i=0; i<n; i++) s+=p[i];
    return s;
}
```

4. # include <iostream.h>

程序运行结果:

```
void main()
{
    int a[ ][4] = {{ {1,2},{3,4},{5,6},{7,8} }};
    int i, j, *p = a[0];
    for(i=0; i<4; i++)
        for(j=0; j<3; j++)
            cout << a[i][j] << " ";
    cout << endl ;
}
```

```

5. #include <iostream.h>

class Base{
    int x;
public :
    Base(int x1){    x=x1;    }
    virtual void result(){    cout<<"Base result is:"<<x*x << endl;    }
};

class Derived : public Base{
    int y;
public :
    Derived(int x1,int y1):Base(x1){    y=y1;    }
    void result(){    cout<<"Derived result is:" <<y*y*y << endl;    }
};

void ShowResult( Base * p ){
    p->result();
}

void main()
{
    Base a(5);
    Derived d1(5,10), d2(10,20);
    ShowResult(&a);
    ShowResult(&d1);
    Base * ptr = &d2;
    ptr->result();
}

```

程序运行结果:

```

6. #include <iostream.h>

class A
{
public:
    virtual void fun(int x)
    {
        cout << x << endl;
    }
};

class B : public A
{
public:
    virtual void fun(int x)
    {
        cout << x*x << endl;
    }
};

void test(A *p)
{
    p->fun(16);
}

void main()
{
    A obj1;
    B obj2;
    test(&obj1);
    test(&obj2);
}

```

程序运行结果:

三、程序填空 (每空 2 分, 共 32 分)

1. 以下程序实现以每行 5 个数输出 300 以内被 4 或 14 整除的偶数, 请将程序补充完整。

```
#include <iostream>
#include <iomanip>
using namespace std;
void main()
{   int i,n;
    _____;
    for(i=1; i<=300; i++)
        if( _____ )
            if(i%2 == 0)
                {   n++;
                    cout<<setw(5)<<i;
                    if( _____ )
                        cout<<endl;
                }
}

```

2. 以下程序实现从键盘输入的 10 个整数中找出最小值并输出到屏幕, 请将程序补充完整。

```
#include <iostream>
using namespace std;
_____;
void main()
{   int d[10], i;
    for(i=0; i<10; i++) cin>>d[i];
    cout<<_____ <<endl;
}

```

```
int findMin(int data[], int size)
{   int i, min;
```

```
for(i=1; i<size; i++)
    _____;
```

```
{   if(data[i]<min) min = data[i];
    _____;
}
```

3. 以下程序的功能是生成斐波那契数列的前 n 项, 直到第 n 项的值大于 2016, 并将这 n 个数存入一维数组 x, 然后将这 n 个数存入文本文件 data.txt。请将程序补充完整。

```
#include <fstream.h>
void main()
{   int x[100], i, n;
    x[0]=0;   x[1]=1;
    _____;
    do{
        x[n]=x[n-1]+x[n-2];
    }while( _____ );
    _____;
    if(!fout) return;
    for(i=0; i<n; i++) fout<<x[i]<<" ";
    fout<<endl;
    fout.close();
}

```

4. 以下程序的功能是: 已知带头节点的单链表, 头指针 head 指向头节点。该单链表中所有节点已按升序排序, 当插入一个新节点时, 要求仍然保持有序。请将程序补充完整。

```
#include <iostream>
using namespace std;
class Node{
public:   int data;
    _____;};
```

学院_____专业_____班_____

年级_____学号_____

姓名_____

共 7 页 第 A5 页

```

class chain{
public:
    chain0{
        head = new Node;
        head->next = NULL;
    }
    ~chain0{ ... } //释放链表内存空间
    //在链表中插入一个新数据
    void Insert(int n){
        Node *p = head,*q;
        _____;
        while(q != NULL && q->data < n)
        {
            _____;
            q = q->next;
        }
        q = new Node; q->data = n;
        q->next = p->next; p->next = q;
    }
    void traverse()
    {
        Node *p;
        p = head->next;
        while(p)
        {
            cout<<p->data<<endl;
            p = p->next;
        }
    }
private:
    Node *head;
};

```

```

void main()
{
    chain c;
    int i, a;
    for(i=0; i<5; i++)
    {
        cin>>a; c.Insert(a);
    }
    c.traverse();
}

```

5. 以下程序实现对运算符“>”进行重载，以实现对两个矩形面积大小的比较。请将程序补充完整。

```

#include <iostream.h>
class rect{
    int length, width;
public:
    rect(int l, int w){ length=l; width=w; }
    _____;
    bool rect:: operator>( rect r){
        _____;
        int a2=r.length*r.width;
        return a1>a2;
    }
    void main(){
        rect r1(5, 5), r2(6, 4);
        if( _____ )
            cout<<"r1 面积大于 r2\n";
        else
            cout<<"r1 面积不大于 r2\n";
    }
}

```

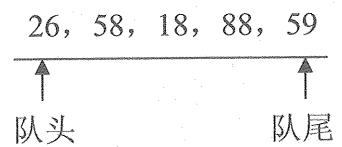
四、简答题 (共 12 分)

1. 给定序列 {7,3,6,8,5,2}, 构造哈夫曼树。(3 分)

3. 如果一个线性表中的元素总数基本稳定, 且很少进行插入和删除操作, 但要求以最快的速度存取表中指定位置的元素, 对该线性表应选用何种存储结构, 为什么? (2 分)

2. 设哈希函数为 $h(key)=key \% 5$, 使用链地址法解决冲突, 画出关键字序列 {23, 45, 71, 12, 55, 74, 35} 的哈希表存储结构。(4 分)

4. 设栈 S 的初始状态为空, 队列 Q 的初始状态为:



对栈 S 和队列 Q 进行以下操作:

- (1) Q 中元素依次出队, 并压入栈 S 中, 直至 Q 为空;
- (2) 依次弹出 S 中的元素并进入 Q, 直至 S 为空。

画出在上述操作后队列 Q 的状态。(3 分)

学院_____专业_____ 班_____ 年级_____ 学号_____ 姓名_____

共 7 页 第 A7 页

五、编写程序（共 12 分）

编写函数 void countDigit(char *p, int count[10]), 其功能是统计字符串 p 中数字 0-9 分别出现的次数，并储存在数组 count 中。编写主函数实现如下功能：从键盘输入长度不超过 79 的字符串（可包含空格），通过调用 countDigit() 函数统计数字 0-9 分别出现的次数，并把结果显示在屏幕上，程序运行示例如图 1 所示。

```
请输入一串字符: sdf jk324jvmb124fmsdf90235ipp89
字符串sdf jk324jvmb124fmsdf90235ipp89中
数字0出现的次数为1
数字1出现的次数为1
数字2出现的次数为3
数字3出现的次数为2
数字4出现的次数为2
数字5出现的次数为1
数字6出现的次数为0
数字7出现的次数为0
数字8出现的次数为1
数字9出现的次数为2
Press any key to continue
```

图 1 程序运行示例

2016~2017 学年第一学期期末考试

《计算机软件技术基础 2》(C++、64 学时) (A 卷) 参考答案

一、单选题，将答案填在下面的表格中，直接标在题上不记分（每小题 1 分，共 20 分）

1. A 2. B 3. C 4. C 5. A 6. D 7. A 8. C 9. A 10. B
11. B 12. C 13. C 14. D 15. D 16. C 17. B 18. C 19. B 20. B

二、写出以下程序的运行结果（每小题 4 分，共 24 分）

1. 10 12

2. 8 4

7 2

6 3

4 2

5

3. 3

15

28

45

4. 1 2 2

3 0 0

0 6 0

7 0 0

5. Base result is:25

Derevid result is:1000

Derevid result is:8000

6. 16

256

三、程序填空（每空 2 分，共 32 分）

1. n = 0 i%4 == 0 || i%14 == 0 n%5 == 0

2. int findMin(int data[], int size) findMin(d,10) min = data[0] return min

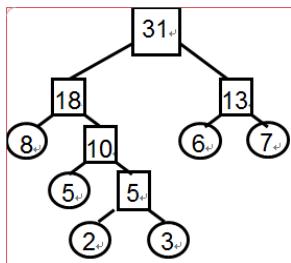
3. n=2 x[n++]<=2016 ofstream fout("data.txt")

4. Node *next q = p->next p = q

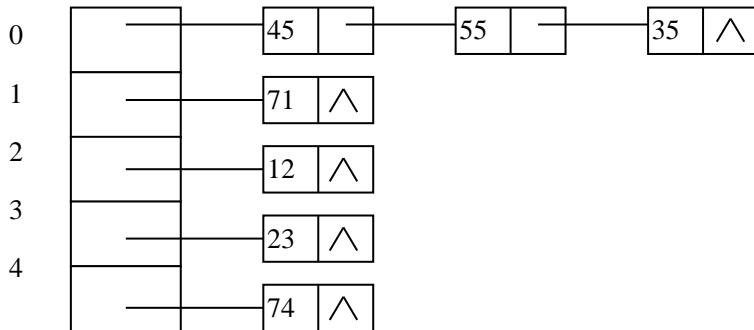
5. bool operator>(rect r) int a1=length*width r1>r2

四、简答题（共 12 分）

1. (3 分)

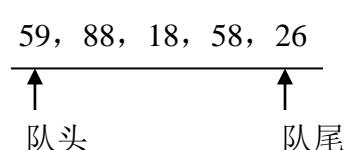


2.



3. 应选用顺序存储结构。因为，在以顺序方式存储线性表时，线性表中的元素的存储顺序与逻辑顺序相一致，根据元素的下标可以很快找到它的存储位置。

4.



五、编写程序（共 12 分）

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
void countDigit(char *p, int count[10])
{
    int num,i;
    //初始化计数器
    for (i=0;i<10;i++)
        count[i]=0;
    //统计各数字出现的次数
    for (;*p!=0;p++)

```

```
{    num=*(p-'0');
    if (num>=0 && num<10) //检查是不是数字
        count[num]++;
}
void main()
{
    char str[80];
    int count[10];
    int num(0),i;
    cout<<"请输入一串字符: ";
    cin.getline(str,80);
    countDigit(str,count);
    //将统计好的数据显示
    cout<<"字符串 "<<str<<"中 "<<endl;
    for (i=0;i<10;i++)
        cout<<"数字 "<<i<<"出现的次数为 "<<count[i]<<endl;
}
```