**2017专硕复试C语言回忆版**

**By 我好方**

*本次专硕考试无编程题，坑的一逼，学硕编程占50分，编程的分值我看了一下大致分配给了读程序和填空和简答，填空真的是痛苦，非得按照他的思路走，还不如让我编个程呢……*

*经过整理，发现大题一部分来源于黄迪明的C实验指导与题解，建议仔细看看该文件。群文件里有扫描版PDF。*

郑重感谢各位提供题目及纠正错误的小伙伴们！

1. **选择题 10个×2分（记得不是9个就是8个原题，历年真题看会就行）**

1.下面程序的输出结果是（ ）

int x = 3;

do{printf(“%d\n”,x -= 2);}while(!(-- x));

**x初始值为3，第一次循环中运行printf函数，参数x -= 2的值为1，输出1，此时x = 1，进行判断!(-- x)，x先自减1，为0，取非为1（真），进行第二次循环，x先减2，输出-2，此时x=-2，判断!(-- x)，x先自减1为-3，取非为0（假），结束循环。所以输出结果为：**

**1**

**-2**

2.下面程序的输出结果是（ ）

void main(){

int a[]={1,7,12,15};

int \*p1=a,\*p2 = p1++;

\*p1 += \*p2++;

printf(“%d %d”,\*p1,\*p2);

}

**首先定义指针p1指向数组a首地址，然后定义指针p2，也指向数组首地址，然后p1自加，也就指向了数组第二个元素。\*p1 += \*p2++;语句先将p2指向的第一个元素的值加到p1指向的第二个元素的值上，也就是第二个元素值为8，然后p2自加，指向第二个元素（\*p2++，\*与++优先级相同，从右自左结合，先与++结合，表示语句执行完后p2指向下一个元素，然后与\*结合，表示p2现在所指向的第一个元素的值），所以最后输出结果为：8 8**

3.下面程序的输出结果是（ ）

int func(int \*p){

return (\*p-- = 3) - 1;

}

void main(){

int arr[]={10,7,5};

int \*p = arr + 1;

printf(“%d”,func(p) + \*p);

}

**p指针首先指向数组arr的第二个元素，首先调用func函数，将实参指针p指向的地址传递给函数形参p，形参执行\*p-- = 3，使得p指向的数组第二个元素值为3，然后形参p自减指向第一个元素，但是实参p不变，还是指向第二个元素，此时func返回的值为3 - 1 = 2，然后2 + \*p，这里实参p指向第二个元素，值为3，所以结果为：5**

4.下面程序的输出结果是（ ）

long fib(int n)

{

if(n > 2) return (fib(n - 1) + fib(n - 2));

else return 2; //16年这里返回1，注意区别

}

void main(){

printf(“%d\n”,fib(3));

}

**调用fib(3)，由于3>2，所以返回fib(2)+fib(1)，fib(2)和fib(1)都返回2，所以最后结果为：4**

5.下面程序的输出结果是（ ）

void main()

{

char c = 48;

int i,mark = 01;

for(i = 0;i < 5;i ++)

{

printf(“%c”,c|mark);

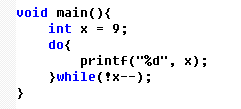
mark = mark << 1;

}

}

**C=48转换成二进制数就是110000，mark初始为八进制1，循环执行5次，每次先输出c|mark（按位或运算）对应的ASCII字符，然后mark左移1位，即乘以2。第一次循环c为110000，mark为1，c|mark为110001，即十进制49，对应ASCII字符‘1’，然后mark左移为10；第二次循环c|mark为110010，即十进制50，对应ASCII字符‘2’，然后mark左移为100，以此类推，最后输出为12480。**

6.



输出结果：9

7.

下面程序的输出结果是（ ）

void main(){

int a,b = 0;

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

for(a = 0;a < 10;a ++)

b += c[a];

printf("%d",b);

}

**程序遍历数组c，将c的每个元素的值累加到b上，所以最后结果为：45**

8.

//这个题有些地方记不清，50 60 70是我随便写的，不过不影响做题，考运算符优先级

struct Stu{

Int num;

Int score;

};

void main(){

struct Stu stu[3] = {{50,50}, {60,60}, {70,70}};

struct Stu \*p;

p = stu[0];

}

问哪个选项是不对的？

选\*p.num = 50，因为.的优先级高于\*，而.是直接取值，改为(\*p).num

1. **指出错误并改正 10分×3个（2、3是以前原题）**

1.

Void main(){

char data[]="There are some mistakes in the program";

char \*point;

char array[30];

int i,length;

length=0;

while(data[1ength]!='\0')

length++;

for(i=0;i<length;i++,point++)

\*point=data[i];

array=point;

printf("%s\n",array);

}

经过分析，我们可以看出程序的目的是将这个字符串输出，原程序中的主要错误是在指

针运算前没有赋初始地址，于是进行以下修改：

#include<stdio.h>

main()

{

char data[]="there are some mistakes in the program";

char \*point;

char array[100];//定义一个数组

int i,length;

length=0;

while(data[length]!='\0')//求字符串的长度

length++;

point=array;//给指针赋首地址，主要修改部分

for(i=0;i<=length;i++,point++)//输出字符串 注意这里是<=，\0也被复制进去了

\*point=data[i];

printf("%s\n",array);

}

2.

Int \*p,\*q;

p=malloc(sizeof(int)\*20);

q=malloc(sizeof(int)\*10);

…

q=p;

…

free(p);

free(q);

答：由于free(p);已经将p所指内存区域释放掉，又q和p都指向同个内存区域，之后free(q);试图释放一个已经被释放的内存区域，会使程序崩溃。此外原来q指向的内存区域由于指针丢失，会导致内存泄露。

修正方法：删除语句q=p;

3.交换两个字符串

Void swap(char \*p, char \*q)

{

Char \*temp;

\*temp=\*p;

\*p=\*q;

\*q=\*temp;

}

答：\*temp、\*p、\*q都是指向的一个字符，所以该函数只实现了交换两个字符串的首字符而已。修正：

void swap(char \*p, char \*q)

{

char temp[100];

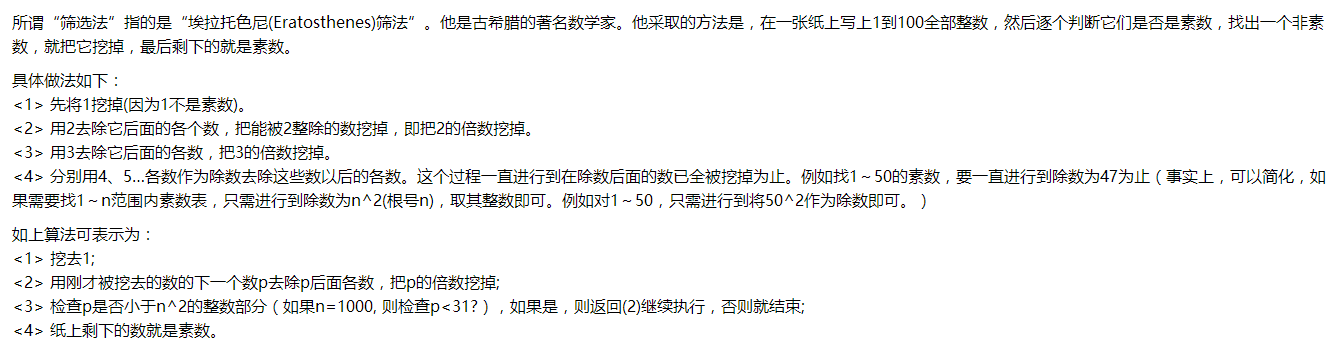
strcpy(temp, p);

strcpy(p, q);

strcpy(q, temp);

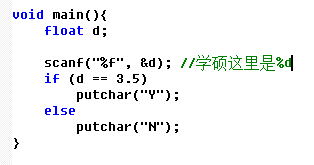
}

1. **简答题**
2. 用Eratoshenes筛选法求素数，好像程序是个错的，让你修改其中三行，提高效率



此题原版程序在群文件里黄迪明教材P137，程序让你修改的是第12、14、15行，其中第12行的d<=sqrt(200)原题给的是d<=200,其余两处修改并不清楚，有清楚的小伙伴欢迎补充！

2.



问：输入3.5，输出结果是什么？为什么？如何修改？

题目问了如何修改，总感觉它想让你回答N，但是验证就是Y，问的怪怪的，不知道出题人怎么想的，不过这只是我的主观臆断……

答：输出Y（经VC6验证过）。因为恰好输入3.5，数位由0补齐，与3.5刚好相等，故输出Y。但是由于浮点数表示都是不精确的，所以不能用“= =”判断相等，有几率出现错误结果。修改方法为if条件变成“d减3.5”的绝对值小于某一精度。

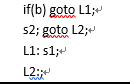
3.C提高效率的方法和原因，年年考，不过专硕的题目是除了使用指针、位运算、宏定义以外还有哪些措施（18分）

4.C语言，除了关键字，还有哪些单词类型（好像是这么问的，百度了一下应该是问的C的语法符号）？C的存储类型关键字有哪四个？

答：（1）标识符，运算符、分隔符、常量、注释符

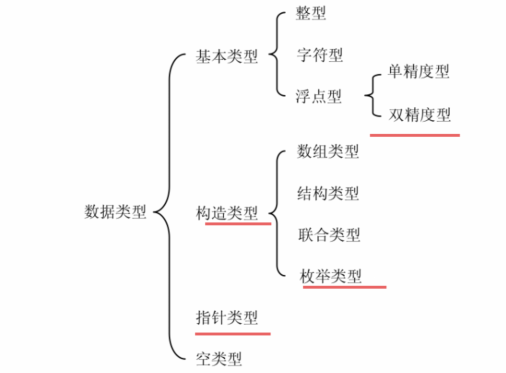
（2）关键字：auto、extern、register、static

5. if(B) S1 else S2 用显示结构表示



6．变量常量填表的那个，题目想不起来了，在15年回忆版中有类似的

7.四个空6分



1. **读程序 6分一个**

读程序确实不好回忆，只能记起来这么多，原题好像有五六个题

1.

void main()

{

char p[10]="UESTC";

char q[]="xyz"; //记不清这是啥了，不影响做题

int i,j;

i=0;

while(\*(p+i)!='\0') i++;

j=0;i++;

while(\*(q+j)!='\0')

{

\*(p+i)=\*(q+j);

j++;

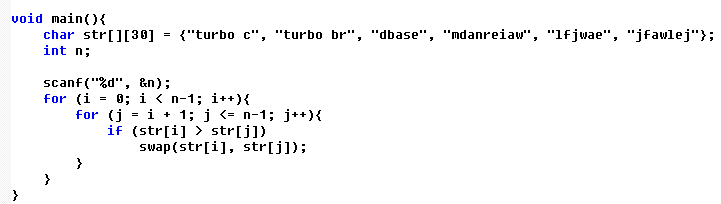
}

printf("%s", p);

}

输出为：UESTC

2.该程序功能是什么？输出结果是啥？题目给的输入n为5。



答：功能：对前n个字符串进行字典排序；结果显而易见

3.读程序的功能

功能：从1-100中找出符合以下条件的数，该数和该数的平方中都含有输入的数字d

#include<stdio.h>

void main()

{

int i,j,k,d,flag1,flag2;

scanf("%d",&d);

printf("d=%d\n", d);

for(i=1;i<=100;i++)

{

j=i;

flag1 = 0;

while((j>0) && (!flag1))

{

k=j%10;

j=j/10;

if(k==d)

flag1=1;

}

if(flag1)

{

j=i\*i;

flag2 = 0;

while((j>0) && (!flag2))

{

k=j%10;

j=j/10 ;

if(k==d)

flag2=1;

}

if(flag2)

printf("%-5d %-5d\n", i, i\*i);

}

}

}

1. **填空题 32个×2分 标红的是空**
2. 快排找第K小（不知道为啥以前的回忆版写的是找K大）

//直接把以前那个的代码扒过来了

Low=0;high=n-1;

do{

I=low;j=high;t=a[i];

do{

While(a[j--]>t);

While(a[i++]<t);

If(i<j) swap(a[j],a[i]);//具体代码实在忘了，反正不是空空，我这样写运行是一样的

}while(i<j);

If(i==k)

Return t;

If(i>k)

high=i-1;

if(i<k)

low=i+1;

}while(low<high);

Return a[low];//这个空我可能错了

1. 在str1中删除出现在str2中的字符，原程序想不起来了，不过是用指针做的。题目的思路就是对于str1中的每一个字符，遍历一遍str2，有则删除。自己编一遍能实现就行~

感谢老习提供的代码，虽然并不是用指针做的，可以当做一个参考

#include<stdio.h>

#include<string.h>

main()

{

char s1[100],s2[100];

int i,j=0,k=0,len;

scanf("%s",s1);

scanf("%s",s2);

len=strlen(s2);

for(i=0;s1[i]!='\0';i++)

{

while(s1[i]!=s2[j])

j++;

if(j>=len)

{

s1[k]=s1[i];//红色标记为填空部分，这个题记不清原代码了，但思想就是我这样。

k++;

}

j=0;

}

s1[k]='\0';

printf("%s",s1);

}

1. 链表逆序，这个题while那个空给的很诡异，！要你自己写

下面给出两种头插法，我想大多数人应该首先想到用王道上讲的头插法，但是王道的头插法用在这里有个弊端，在程序中已标注。

方法一：王道头插法，从head->next开始头插

void reverse(List \*Head){

List \*P = Head->next;

List \*Q;

Head->next = NULL //原题缺这一句，造成尾节点next不为空，无限输出

while(P!=NULL){

Q = P->next;

P->next = Head->next;

Head->next = P;

P = Q;

}

}

方法二：从第三个节点开始头插（第一个节点为head->next）

void reverse(Node \*head){

Node \*p = head->next;

Node \*q;

while(p->next != NULL){

//提供者说while条件里还应该有P不能为空，防止链表为空，大家知道行了，考试就这么给的空，鬼知道要你填几个条件，随机应变好了~~

q = p->next;

p->next = q->next ;

q->next = head->next;

head->next = q;

}

}

1. 如果有两个数，每一个数它的所有约数（除了它本身之外）的和正好等于对方，则称这两个数为互满数，求出30000以内所有的互满数，并输出显示。求互满数函数原型

（int factor(int j)）

题目并没有说求多少以内的，求的话换一下#define N就行

#include<stdio.h>

int k[40];

#define N 30000

int factor(int j)

{

int i,n,s;

n = 1;

s = j;

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

{

if((j%i)==0)

{

n++;

s=si;

k[n]=i; // 将每个因子赋给数组K

}

}

s=1;

for (i=1;i<=n;i++)s+=k[i]; // 将每个因子累加

return s;

}

void main()

{

int s1,s2;

for (int m=2;m<N;m++)

{

s1=factor(m);

for (int n=m+1;n<N;n++)

{

s2=factor(n);

if(s1==s2)printf("%d和%d是互满数\n",m,n);

}

}

}

1. 大概的意思就是输入一个字符串，将连续相同的字符删减为一个，以“\*”结尾

void main(){

char ch\_old, ch\_new;

ch\_old = '\*';

do{

scanf("%c", &ch\_new);

if (ch\_old == ch\_new)

continue;

ch\_old = ch\_new;

printf("%c", ch\_old);

}while(ch\_new != '\*');

}