**一、阅读程序题**（36分，每小题6分）

|  |  |
| --- | --- |
| **得分** |  |

阅读程序，按格式将程序运行的输出结果填写在右边空白区。

1. **#include<stdio.h>**

**void main()**

**{**

**int i, a[]={2,3,5,4};**

**for(i=0;i<4;i++)**

**switch(i%2)**

**{ case 0: switch(a[i]%2)**

**{ case 0: a[i]++; break;**

**case 1: a[i]--;**

**} break;**

**case 1: a[i]=0;**

**}**

**for (i=0; i<4; i++) printf(“%d”,a[i]); printf(“\n”);**

**}**

**Break可以跳出循环和switch语句，谁离得近先跳谁**

1. **#include<stdio.h>**

**void main()**

**{**

**int x,a=1,b=1;**

**while(1)**

**{ scanf(“%d”, &x);**

**if (x>0) {a\*=x; break;}**

**if (x<0) {b\*=x; continue;}**

**printf(“%d,%d\n”,a,b);**

**}**

**}**

**执行时输入下面5个数：-1 -2 0 1 2<回车>后，则输出结果是：**

**Continue跳出单次循环，进入下一次循环**

**Break跳出整个循环**

1. **#include <stdio.h>**

**int f(int n)**

**{**

**static int a=1;**

**n=a++;**

**return n;**

**}**

**void main()**

**{**

**int a=3, s;**

**s=f(a); s=s+f(a); printf(“%d\n”,s);**

**}**

**主函数：**

**第一次调用s=f(3);**

**n=3;(n的赋值没有用，就是迷惑人的)**

**a=1;(a是静态，只会在第一次赋值，后面直接调用，不会再赋值)**

**n=a++;从左到右，先算n=a=1，再算a++;a=2;返回值s=2;**

**(如果是n=++a;就是先算++a;a=2;再算n=a=2;因为自增运算符优先级高于赋值运算符)**

**第二次调用s=2+f（3）；**

**n=3;(没有用)**

**a=2;(静态a，直接调用上一次用的a的值)**

**n=a++;**

**n=2;a=3;**

**返回值s=1+2;**

1. **#include <stdio.h>**

**#include <string.h>**

**void fun(char \*s,int m1, int m2)**

**{ char t,\*p;**

**p=s+m1; s=s+m2;**

**while (s<p)**

**{ t=\*s; \*s=\*p; \*p=t;**

**s++; p--;**

**}**

**}**

**void main()**

**{**

**char ss[10]=“876543210”;**

**int n=6;**

**fun(ss,0,n-1);**

**fun(ss,9,n);**

**fun(ss,0,9);**

**printf(“%s\n”,ss);**

**}**

**初始：（黑色是位置/地址，红色是数值）**

**0 1 2 3 4 5 6 7 8 9**

**8 7 6 5 4 3 2 1 0 \0**

**第一次调用函数**

**M1=0,m2=5**

**P=0;(指向字符串第0位)**

**S=5；(指向字符串第5位)**

**S，p指向地址，\*s，\*p指向的是字符**

**s>p 所以第一次调用结束，返回主函数**

**第二次调用函数**

**M1=9，m2=6；**

**P=9,s=6**

**p>s 进入循环**

**交换第6位和第9位**

**0 1 2 3 4 5 6 7 8 9**

**8 7 6 5 4 3 \0 1 0 2**

**p--;p=8;**

**s++;s=7;**

**交换第7位和第8位**

**0 1 2 3 4 5 6 7 8 9**

**8 7 6 5 4 3 \0 0 1 2**

**第三次调用函数**

**M1=0，m2=9；**

**P=0,s=9**

**s>p 所以第一次调用结束，返回主函数**

**\0是结束字符串的标准，输出到\0就结束了**

**所以输出到第6位就结束了：876543**

**如果把字符串改为数组**

**第9位自动补0**

**交换之后输出8765430012**

5. **#include <stdio.h>**

**struct T {int x; struct T \*y;}data[2]={10, 0, 20, 0};**

**void main()**

**{ struct T \*p=data;**

**p->y=data;**

**printf("％d\n",++p->x);**

**}**

**struct T \*y；是指y是结构体struct的地址**

**初始化data[0].x=10;data[1].x=20;**

**\*y=0;是空指针，后面赋值**

**struct T \*p=data;赋值**

**下一句没看出多大用，删了程序一样跑**

**输出先p->x，就是指向data[0].x=10;**

**再++x，输出11 （如果是x++，那就是先输出10，再加1）**

**结构体运算符优先级高于自增运算符（书附录c P359）**

6. **#include <stdio.h>**

**void main()**

**{ int a[3][4]={2, 4, 6, 8, 10, 12,14,16,18,20,22,24};**

**//假设数组a的地址以十进制表示，首地址为2000，每个整数占4个字节**

**printf("%d\n", a);**

**printf("%d\n", \*a+2);**

**printf("%d\n", \*(\*(a+2)+2));**

**}**

**2000 2004 2008 2012**

**2 4 6 8**

**2016 2020 2024 2028**

**10 12 14 16**

**2032 2036 2040 2044**

**18 20 22 24**

**a是a[0][0]=2的地址2000**

**\*a是第0行a[0]的地址，默认是第0个数a[0][0]的地址，\*a+2是第0行第2个数a[0][2]=6的地址2008**

**//如果是\*(a+1)或者a+1,就是第1行a[1]的地址,默认是第0个数a[1][0]的地址**

**//如果是\*(a+1)+2,就是第1行第2个数a[1][2]的地址**

**所以\*(a+2)+2就是第2行第2个数a[2][2]=22的地址**

**再加一个\*(\*(a+2)+2)就是第2行第2个数a[2][2]=22**

**书P245**

|  |  |
| --- | --- |
| **得分** |  |

**二、程序分析题**（24分，每小题6分）

**请将下列程序补充完整，填写在相应的横线内。**

1. 输入10个整数，用冒泡法对它们按由小到大的顺序排序。

|  |
| --- |
|  |
|  |
|  |

**#include <stdio.h>**

**void main()**

**{ int i,j,k,a[10];**

**for(i=0;i<10;i++)**

**scanf(“%d”,&a[i]);**

**for(i=0;i<9 ;i++)**

**for(j=0;j ;j++)**

**if( )**

**{k=a[j];**

**;**

**;**

**}**

**for(i=0;i<=9;i++)**

**printf("%5d",a[i]);**

**printf("\n");**

**}**

**冒泡排序 书P144**

2.下列给定程序中，函数fun的功能是：将形参s所指字符串中的所有字母字符顺序前移，其他字符顺序后移，处理后将新字符串的首地址作为函数值返回。例如，若s所指字符串为“fpg234symn687tr”,处理后新字符串为“fpgsymntr234687”。

**#include <stdio.h>**

**#include <string.h>**

**#include <stdlib.h>**

**char \*fun(char \*s)**

**{ int i,j,k,n; char \*p, \*t;**

**n=strlen(s) +1;**

**t=(char \*)malloc(n\*sizeof(char));**

**p=(char \*)malloc(n\*sizeof(char));**

**j=0; k=0;**

**for （i=0; i<n; i++）**

**{if (((s[i]>=’a’)&&(s[i]<=’z’))|| ((s[i]>=’A’)&&(s[i]<=’Z’)))**

**{ t[j]= ; j++;}**

**else {p[k]=s[i]; k++;}**

**}**

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

**t[j+i]=p[i];**

**t[j+k]= ;//循环已经结束，这是最后一个空，字符串末尾记得有’\0’**

**return t;**

**}**

**void main()**

**{ char s[80];**

**printf(“please input:”);**

**scanf(“%s”,s);**

**printf(“\nThe result is : %s\n”,fun(s));**

**}**