# **指针3课后作业**

**一、若指针p已正确定义，要使p指向两个连续的整型动态存储单元，不正确的语句是（A）。**

**A、p=2\*(int\*)malloc(sizeof(int));**

**B、p=(int\*)malloc(2\*sizeof(int));**

**C、p=(int\*)ma1loc(2\*4);**

**D、p=(int\*)calloc(2,sizeof(int));**

**二、若定义了以下函数:**

**void f(……){**

**……**

**\*p=(double\*)malloc(10\*sizeof(double));**

**……**

**}**

**p是该函数的形参，要求通过p把动态分配存储单元的地址传回主调函数，则形参p的正确定义应当是（C）。**

**A、double \*p**

**B、double (\*p)[10]**

**C、double \*\*p**

**D、float \*\*p**

**三、如有以下定义：**

**typedef struct Teacher{ //教师结构体**

**int Tid;**

**char Tname[20];**

**char Tsex[4];**

**int Tage;**

**}T;**

**typedef struct Class{ //班级结构体**

**int Cid;**

**int Cname[20];**

**T \*C\_T; //教师结构体指针成员变量**

**int num = 0;**

**}C;**

**typedef struct Student{ //学生结构体**

**int Sid;**

**char Sname[20];**

**char Ssex[4];**

**int Sage;**

**C \*S\_C; //教师结构体指针成员变量**

**float score;**

**}S;**

**T t1[3];**

**C c1[3];**

**S s1[10],\*Sp;**

**Sp=s1;**

**以下能够表示学生3所在班级的教师名称的选项是（B）。**

**A、Sp[3].S\_C->C\_T->Tname**

**B、(Sp+2)->S\_C->C\_T->Tname**

**C、Sp[2]->S\_C->C\_T->Tname**

**D、Sp[3].S\_C.C\_T.Tname**

**四、如有以下程序段：**

**#include <stdlib.h>**

**main()**

**{**

**char \*p,\*q;**

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

**q=p;**

**scanf("%s%s", p, q) ;**

**printf("%s%s\n", p, q);**

**}**

**若从键盘输入: abc def（回车），则输出结果是（D）。**

**A、def def**

**B、abc def**

**C、abcdef**

**D、defdef**

**五、如有以下程序**

**#include <stdio.h>**

**#include <stdlib.h>**

**void fun(int \*p1,int \*p2,int \*s)**

**{**

**s=(int\*)malloc(sizeof(int)) ;**

**\*s=\*p1+\*p2;**

**free(s);**

**}**

**void main()**

**{**

**int a=1,b=40,\*s=&a;**

**fun (&a,&b, s);**

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

**}**

**程序运行后的输出结果是（C）。**

**A、42**

**B、0**

**C、1**

**D、41**

**六、（多选）以下关于内存四区的说法中正确的是（ABCD）。**

**A、栈区：存放函数的形参、局部变量等。由编译器自动分配和释放，当函数执行完毕时自动释放。**

**B、堆区：用于动态内存的申请与释放，一般由程序员手动分配和释放，若程序员不释放，则程序结束时由操作系统回收。**

**C、全局静态常量区（全局区）：存放常量（一般是字符串常量和其他常量）、全局变量和静态变量，在程序结束后由操作系统释放。**

**D、代码区：存放可执行的代码，一般为CPU 执行的机器指令。**

**七、如有以下程序：**

**typedef struct test**

**{**

**int x, y;**

**}\*T;**

**void main()**

**{**

**T t1, t2, t3;**

**t1 = t2 = t3 = (T)malloc(sizeof(struct test));**

**t1->x = t1->y = 1;**

**t2->x = t2->y = 2;**

**t3->x = t3->y = 3;**

**printf("%d %d\n", t1->x, (\*t2).y);**

**}**

**其输出结果为（）。3 3**

**八、如有以下程序：**

**#define N 3**

**typedef int TYPE;**

**void test1(TYPE \*p,TYPE \*\*q)**

**{**

**\*q = (TYPE \*)calloc(N, sizeof(TYPE));**

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

**{**

**(\*q)[i] = \*p+\*p++;**

**}**

**}**

**void main()**

**{**

**TYPE arr[N] = { 1, 2, 3 },\*r;**

**test1(arr, &r);**

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

**{**

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

**}**

**}**

**其运行结果为（）。2,4,6,**

**九、编写程序，实现以下功能：**

**①声明一个矩形结构体类型，存储矩形相关信息(位置（x坐标，y坐标），长，宽)。**

**②编写函数，计算矩形的周长并将周长返回。**

**③编写函数，计算矩形的面积并将面积返回。**

**④编写函数，判断两个矩形是否相交，如果相交则返回1，否则返回0。**

**（注意：以上功能用结构体指针实现）**

**十、编写程序，实现以下功能：**

**某火车有10节车厢，为了方便管理，需要记录每节车厢中的乘客人数和乘务员人数。**

**①使用结构体记录每节车厢的乘客和乘务员人数。**

**②每节车厢之间用结构体指针连接起来。**

**③编写函数，运用结构体指针从车头开始，遍历整辆火车，分别输入每节车厢中乘客和乘务员的人数。**

**④编写函数，运用结构体指针从车头开始，遍历整辆火车，分别输出每节车厢中乘客和乘务员的人数。**

**⑤编写函数，运用结构体指针，分别统计整辆火车中乘客和乘务员的人数并运用指针返回。**