**2019级电信学院《计算机与程序设计基础(C)》考试试题（A）**

学院\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 班级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 学号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 成绩\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

密

封

线

（考试时间：2020.01.06 考试时长: 3小时）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **题目编号** | **1**(10’) | **2**(5’) | **3**(10’) | **4**(15’) | **5**(20’) | **6**(15’) | **7**(25’) | **总分**(100’) |
| **得分** |  |  |  |  |  |  |  |  |

### 一、判断下列语句或程序段的对错。（“×”表示错，“√”表示对）（10分）

* 1. int x = y = z = 100; ( )
  2. int number = ( 0x18 == 'A' ); ( )
  3. float number = 3.92E-0.9; ( )
  4. char array[] = {1, 2, 3, 4, 5, 6}; ( )
  5. int iArray[2][] = {10, 20, 30, 40, 50}; ( )
  6. char str[20];

str = "string"; ( )

* 1. int a = 5, b = 7, c;

执行c = a+++(++b) 语句后，a、b、c的值分别为6, 8, 12。 ( )

* 1. char \*chPtr = NULL;

scanf( "%s", chPtr ); ( )

* 1. int (\*aPtr)[8], array[6][8];

aPtr=array+1; ( )

* 1. register int n;

scanf( "%d", &n ); ( )

### 二、计算表达式的值 (假定各题计算彼此独立) （5分）

设**unsigned char** x=3, y=5, a=7, b=14, c=6, d=8; **float** f=1.0; **char result**;

* 1. **f = b/x + d%y;** //求f的值 ( )
  2. **!(x > y) && (++a - b)** //求表达式值 ( )
  3. **result = (a & b) ^ ( ~c | d);** //求result的值 ( )
  4. **a += b %= a+b;**  //求a的值 ( )
  5. **a++, a - d;** //求运算结果 ( )

### 三、单选题（10分）

* 1. 设整型变量a、b、c的值均为2，运行表达式语句“++a && ( ++ b || ++c)”后c的值是( )

A) 3 B) 2 C) 1 D) 0

* 1. 设char strArray[15] = {"Aa\t12\n\\**\0**is15\n"};则sizeof(strArray)的值是（ ）

A) 14 B) 15 C)7 D) 8

* 1. char型常量在内存中存入的是( )

A) ASCII代码值 B) BCD代码值 C) 内码值 D) 十进制代码值

* 1. 设有float f = 13.8; int num; 执行语句 **num = ( (int) f )/3**; 后，变量num的值是( )

A) 1 B) 4 C) 4.333333 D) 4.6

* 1. C语言规定，函数返回值的数据类型是由( )

A) return语句中的表达式类型所决定

B) 调用该函数时的主调函数类型所决定

C) 调用该函数时系统临时决定

D) 在定义该函数时所指定的函数类型所决定

* 1. 在C语言中，数据变量的**缺省存储类型**是( )

A) auto B) static C) extern D) register

* 1. 设str是一个字符串数组( **char str[80];** )，下面哪一条语句的执行结果和其他语句的执行结果是**不同的**( )

A) \*str=0; B) str[0]= '\0'; C) strcpy(str, ""); D) strcpy(str, "0");

* 1. 设有如下变量定义: **int a[ ]= {1, 3, 5, 7, 9, 11, 13}, x, \*p=a+3;** 则下列哪一条语句能够将数组中的**数值5**正确赋值给变量**x** ( )

A) x= \*p++ B) x = \* ( --p ) C) x = \*( ++p ) D) x = \*( p-- )

* 1. 下列代码段中，**合法且逻辑正确**的代码选项是( )

A) **float** f, \*fp= &f; B) **char** a[10]= "SampleString";

C) **char** \*t; scanf("%s", t); D) **int** a[2][ ]={{1, 2}, {3, 4, 7}};

* 1. 在C语言中，假定一维数组作为函数调用时的实参，则实际上在函数调用之时传递给函数的形参是( )

A) 一维数组的所有元素值 B) 一维数组第一个元素的值

C) 一维数组第一个元素的地址 D) 以上都不对

### 四、输出代码程序块运行结果（假定所需的头文件已经包括）（15分）

* 1. **程序代码块(3’)**

|  |
| --- |
| **void main( ){**  **char array[30];**  **// copy strings to the array**  **strcpy(&array[0], "CH");**  **strcpy(&array[2], "IBC");**  **strcpy(&array[3], "NA!");**  **printf("%s\n",array);**  **}** |

**程序结果：**

* 1. **程序代码块(3’)**

|  |
| --- |
| **void strFunc(char\* pstr){**  **int len = strlen(pstr);**  **for(int i=len-1; i>=0; i--){**  **printf("%c ", \*(pstr+i) );**  **}**  **}**  **void main(){**  **char strArray[20] = "Hello,World!";**  **strFunc(strArray);**  **}** |

**程序结果：**

* 1. **程序代码块(3’)**

|  |
| --- |
| **int glbCnt; /\* global var \*/**  **void func(){**  **static int stcVar=5; /\* static var \*/**  **int num=5; /\* default storage type \*/**  **printf("num=%d, stcVar=%d, glbCnt=%d\n",**  **--num,--stcVar, glbCnt++);**  **}**  **void main(){**  **// run the func for 5 times**  **for(int i=0; i<5; i++){**  **func();**  **}**  **}** |

**程序结果：**

* 1. **程序代码块(3’)**

|  |
| --- |
| **typedef struct Key{**  **char \*keyword;**  **int keyno;**  **}Map;**  **void main(){**  **Map set[] = { {"design",123},{"module",456},**  **{"test", 789} };**  **Map \*mPtr;**  **int id;**  **char \*strPtr;**  **mPtr = set; /\* get the map array header \*/**  **id = mPtr->keyno;**  **strPtr = mPtr->keyword;**  **printf("%s, %d\n", strPtr, id);**  **id = (++mPtr)->keyno;**  **strPtr = (mPtr++)->keyword;**  **printf("%s, %d\n", strPtr, id);**  **id = ++mPtr->keyno;**  **strPtr = ++mPtr->keyword;**  **printf("%s, %d\n", strPtr, id);**  **}** |

**程序结果：**

* 1. **程序代码块(3’)**

|  |
| --- |
| **#define SIZE 10**  **int recursionFunc(int b[], int len ); /\* prototype \*/**  **void main(void){**  **int result;**  **int array[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};**  **result = recursionFunc(array, SIZE);**  **printf("result is %d\n", result);**  **} /\* end main \*/**  **int recursionFunc(int b[], int len ){**    **if( len==1 ){ /\* base case \*/**  **return b[0];**  **}**  **else{ /\* recursion step \*/**  **return b[len-1] + recursionFunc(b, len-1);**  **}**  **} /\* end function recursionFunc \*/** |

**程序结果：**

### 五、程序改错（20分）

**题目要求：不得改变程序框架，不得重写程序，无需文字说明，可以直接在代码上添加、删除和修改，也可以将结果填写在有“代码修正”的位置处。**

1. **从键盘获得两个整数，输出较小的整数。(2’)**

|  |  |
| --- | --- |
| ***源代码*** | ***代码修正*** |
| **int x, y;** |  |
| **scanf("%d%d", x, y);** |  |
| **printf("the min is %d\n, (x>y)?x:y" );** |  |

1. **对10个元素数组进行初始化，数组元素值域为[1, 3, 5, 7, … 19]。(3’)**

|  |  |
| --- | --- |
| ***源代码*** | ***代码修正*** |
| **#define SIZE=10** |  |
| **void main( void ) {** |  |
| **int a[SIZE] = {0}, i;** |  |
| **for ( i = 1, i <= SIZE, i++ )** |  |
| **a[i] = 2 \* (i-1);** |  |
| **return 0;** |  |
| **}** |  |

1. **函数计算三个整数的平方值与立方值，需要将结果反馈给主调函数。(5’)**

|  |  |
| --- | --- |
| ***源代码*** | ***代码修正*** |
| **void calculate(int x, int y, int sqr){** |  |
| **int \*temp;** |  |
| **sqr = x\*x + y\*y + z\*z;** |  |
| **\*temp = x\*x\*x + y\*y\*y + z\*z\*z;** |  |
| **return \*temp;** |  |
| **}** |  |

1. **函数void swap (char \*, char \*, int)实现两个长度相同字符串的内容交换。（4’）**

|  |  |
| --- | --- |
| ***源代码*** | ***代码修正*** |
| **void swap(char \*pstr1,char \*pstr2,int len){** |  |
| **char \*tmp;** |  |
| **tmp = pstr1;** |  |
| **pstr1 = pstr2;** |  |
| **pstr2 = tmp;** |  |
| **}** |  |

1. **将键盘输入的字符串进行大小写转换，小写字符转化为大写字符，大写字符转化为小写字符，最后将转换后的字符串进行输出。（6’）**

|  |
| --- |
| ***源代码：*（此题可直接将修改结果放置在卷子对应空白处）** |
| **int main(){**    **char chArray[SIZE] = {0};**  **printf("Please input a string: \n");**  **scanf("%100s", chArray);**    **charConversion(chArray[]);**    **printf("\nThe converted string: \n");**  **printf("%s\n", chArray[0]);**  **}**  **void charConversion(char array[]){**    **int i=0;**    **while(array[i]){**    **if( isUpperCase(array[i]) ){**    **array[i] = array[i] - ('a'-'A');**  **}**  **if( isLowerCase(array[i]) ){**    **array[i] = array[i] + ('a'-'A');**    **}**  **} /\* end of while \*/**  **} /\* end of charConversion \*/**  **int isUpperCase(char ch){**  **return (ch>'A'&&ch<'Z');**  **}**  **int isLowerCase(char ch){**  **return (ch>'a'&&ch<'z');**  **}** |

### 六、程序填空（15）

1. 完善函数功能，函数描述：**计算整型数组所有元素的和。(3’)**

|  |
| --- |
| **int sumArray(int array[], int size){**  **int i;**  **int result = 0;**  **for( #1 ){**  **result = #2 ;**  **}**  **#3**  **}** |

**#1:**

**#2:**

**#3:**

1. **已知斐波那契数列**：0, 1, 1, 2, 3, 5, 8, 13, 21, …定义fibonacci(0) = 0; fibonacci(1) = 1; fibonacci(n) = fibonacci(n – 1) + fibonacci(n – 2);**利用递归方法实现**fibonacci函数**。(3’)**

|  |
| --- |
| **int fibonacci(int n){**  **if( #1 )**  **return 0;**  **else if( #2 )**  **return 1;**  **else**  **return #3 ;**  **}** |

**#1:**

**#2:**

**#3:**

1. 完成程序功能，要求能够正确输出二维数组元素地址及对应值，使用**指针+偏移量**的访问方法实现。**(3’)**

|  |
| --- |
| **void main(){**  **int x[2][3] = {10, 20, 30, 40, 50, 60};**  **int i = 0, j = 0;**  **int #1 ;**  **for(i=0; i<2; i++){**  **for(int j=0; j<3; j++){**  **printf("[%p]:%d \t", #2 , #3 );**  **}**  **printf("\n");**  **}**  **}** |

**#1:**

**#2:**

**#3:**

1. 完善函数功能，函数描述：整型数组的冒泡排序，按照升序从小到大排序。**(3’)**

|  |
| --- |
| **void bubbleSort(int x[], int xSize){**  **int i = 0, j = 0, temp = 0;**    **for( #1 ){**  **for( #2 ){**  **if( #3 ){**    **temp = x[j];**  **x[j] = x[j+1]; ;**  **x[j+1] = temp;**  **} // end of if**  **} // end of inner for**  **} // end of outer for**  **} // end of func** |

**#1:**

**#2:**

**#3:**

1. 完善函数功能，函数描述：利用**迭代方法**实现折半查找(二分查找)算法，**若没找到返回-1**。**(3’)**

|  |
| --- |
| **int binarySearch (int array[],int size,int key){**  **int low = 0;**  **int high = size-1;**  **int mid;**  **while( #1 ){**  **mid = (low+high)/2;**  **if(array[mid]==key){**  **#2 ;**  **}**  **else if(array[mid]>key){**  **high = mid-1;**  **}**  **else{// array[mid]<key**  **#3 ;**  **}**  **} // end of while control**  **return -1; // no found**  **}** |

**#1:**

**#2:**

**#3:**

### 六、编写程序（25分, 6’+7’+12’）

**题目要求：根据题目需要完成代码编写，不得使用全局变量，不得使用goto语句，字迹工整，逻辑清晰，程序结构合理。两个标准头文件stdio.h和stdlib.h勿用复写，其他头文件若使用，需要自行增加到代码中。可以使用宏定义。**

1. 编写程序：提供两个整数的输入，并计算其**最大公约数**和**最小公倍数**，并将结果输出。要求**最大公约数以函数的形式提供**。(6’)
2. 编写函数**myStringFunc**实现两个字符串交替复制功能，比如字符串1为"ac024"，字符串2为"bd135"，调用该函数后，会生成一个**新的字符串**，新字符串的内容为"abcd012345"。**参考测试样例：**字符串1为"ac024"，字符串2为"bd"，新字符串为"abcd024"；字符串1为"1234"，字符串2为"abcdef"，新字符串为"1a2b3c4def"。**myStringFunc**要求：**1. 函数返回值**：**新字符串首地址**；**2：函数参数：**提供两个字符串作为参数，要求函数对该字符串参数不能修改。

请给出**myStringFunc原型声明**以及**具体实现，编写主函数进行测试**。(7’)

1. 设计一个公司的工资管理程序，需要记录公司员工的姓名，性别，工龄，工资，以及**工资发放币种组合**（工资为整数，发放的工资各种钞票限定为100元，50元，20元，10元，5元，1元，发放的钞票数张数要求为最少），定义如下数据结构体**Worker**，用来记录以上信息：

|  |
| --- |
| **#define LENGTH 20**  **typedef struct worker{**  **char name[LENGTH];**  **// M:male(男), F:female(女)**  **char gender;**  **int workAge;**  **int wage;**  **//记录不同币种的数量: 参考[0]放置100元的数量……**  **//参考:[0]:100,[1]:50,[2]:20,[3]:10,[4]:5,[5]:1**  **int combi[6];**  **}Worker;** |

* **完成以下函数的编写：**

1. 提供**void getWorkerInfo(Worker\* worker)函数**，记录**一个员工的信息**，并通过worker返回给主调函数。**(3’)**

2. 提供**void sortWorkersByWage(Worker\* workers, int len)**函数,对员工信息结构体数组进行排序，排序条件为**员工工资条件（由高到低）**。**(3’)**

3. 提供**void displaySingleWorker(Worker worker)**函数，能够对一个员工的信息进行完整的显示，显示顺序为姓名、性别、工龄、工资、工资发放组合，一个具体示例如下：**Tom,M,10,8585,[100:85 50:1 20:1 10:1 5:1 1:0]**。**(2’)**

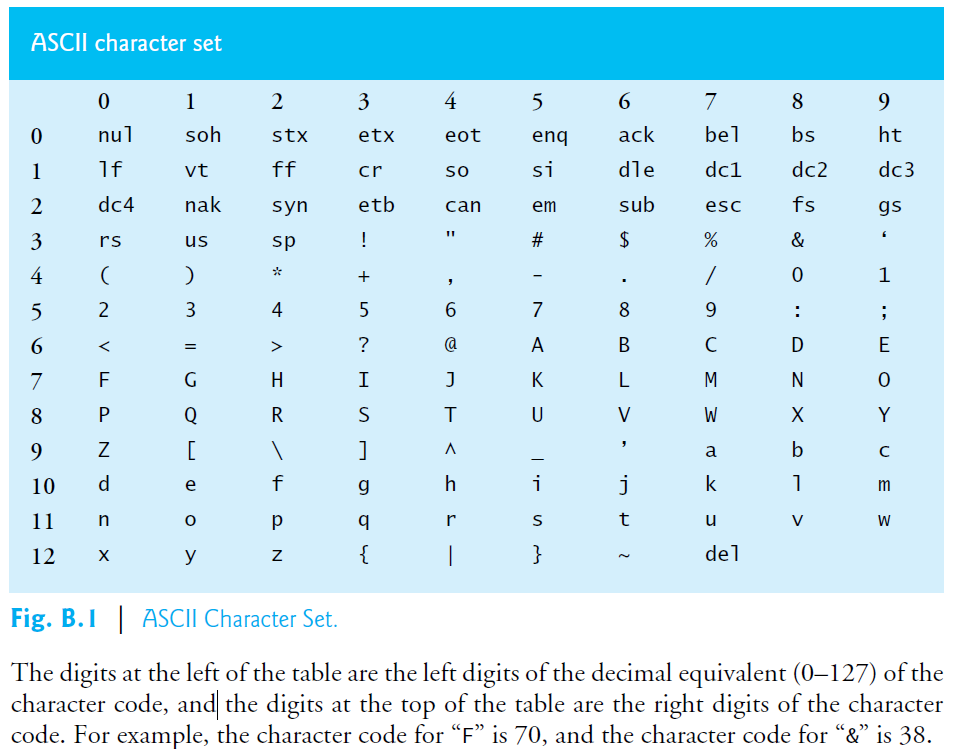
4.提供**void displayWorkers(Worker\* workers, int len)**函数，通过**displaySingleWorker**对员工信息结构体数组进行统一输出显示。**(2’)**

* **测试主程序要求：**

1. 实现并使用定义的函数，要求能够读入5个员工信息；对员工信息根据工资进行排序，显示所有员工的信息；最后输出工龄大于10年，工资高于5000元的所有女员工信息。**(2’)**

### 附录：

* **ASCII字符表**



* **常用函数表**

|  |  |
| --- | --- |
| **函数原型** | **函数描述** |
| **char \*strcpy( char \*s1, const char \*s2 )** | |
| 描述： | 拷贝字符串s2内容到字符数组s1中。返回值为s1。 |
| **char \*strcat( char \*s1, const char \*s2 )** | |
| 描述： | 附加字符串s2内容到字符数组s1中。返回值为s1。 |
| **int strcmp( const char \*s1, const char \*s2 );** | |
| 描述： | 比较字符串s1和字符串s2。 函数返回值0, 小于0或者大于0，分别代表s1等于, 小于或者大于s2。 |
| **void srand(unsigned int seed)；** | |
| 描述 | 随机数发生器的初始化函数。 |
| **int rand( void );** | |
| 描述： | 该函数返回一个在0和RAND\_MAX之间的伪随机数。 |
| **void \*malloc( size\_t size );** | |
| 描述： | 该函数指向大小为size内存的指针，如果失败返回NULL。 |
| **void free( void\* ptr );** | |
| 描述： | 该函数释放指针所指向的内存区域。 |