# 《C语言开发入门教程》习题答案

# 第1章 初识C语言

1. 填空题
2. 汇编语言、高级语言
3. .c
4. Stadio.h
5. return
6. 编译、链接
7. 判断题

1、× 2、√ 3、× 4、√ 5、√

1. 选择题

1、A 2、B 3、C、D 4、D 5、B

四、简答题

1、printf()，该函数用于输出一行信息，可以简单理解为向控制台输出文字或符号等。printf()括号中的内容称为函数的参数，括号内可以看到输出的字符串“Hello, world\n”，其中“\n”表示换行操作，它不会输出到控制台。

2、注释是对程序的某个功能或者某行代码的解释说明，它只在C语言源文件中有效，在编译时会被编译器忽略。

五、编程题

1、参考答案

#include<stdio.h>

void main()

{

printf("我喜欢C语言！");

getchar(); //等待输入一个字符，起到让运行结果界面暂停

}

# 第2章 数据类型与运算符

1. 填空题
2. 0、0x/0X
3. 下划线
4. 原码、反码、补码
5. 构造类型、空类型
6. 获取数据字节数
7. 判断题

1、× 2、× 3、√ 4、× 5、√

1. 选择题

1、A、B、D 2、B 3、B 4、C 5、D

四、简答题

1、在C语言中根据运算符进行运算需根据变量的个数可以将运算符分为单目运算符、双目运算符和三目运算符，其区别与示例如下表：

|  |  |  |
| --- | --- | --- |
| **名称** | **运算所需变量个数** | **范例** |
| 单目运算符 | 1个 | ++、--、!、sizeof、~、… |
| 双目运算符 | 2个 | +、-、\*、/、%、<<、==、… |
| 三目运算符 | 3个 | ?: |

2、在进行自增（++）和自减（--）的运算时，如果运算符（++或--）放在操作数的前面则是先进行自增或自减运算，再进行其他运算。反之，如果运算符放在操作数的后面则是先进行其他运算再进行自增或自减运算。

五、编程题

1、参考答案

#include<stdio.h>

void main()

{

float a = 1.5, b = 2.6, h = 5;

float s;

s = (a + b)\*h;

printf("梯形的面积为：%.2f\n", s); //%.2f输出小数点后2位数字

getchar();

}

2、参考答案

#include<stdio.h>

void main()

{

int a = 3, b = 4;

a = a^b;

b = b^a;

a = a^b;

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

getchar();

}

# 第3章 流程控制

1. 填空题
2. break
3. 顺序
4. 循环
5. if、if…else、if…else if…else
6. 3
7. 判断题

1、× 2、× 3、√ 4、× 5、√

1. 选择题

1、C 2、A、C 3、B、C、D 4、D 5、A

四、简答题

1、跳转语句

* break当它出现在switch条件语句中时，作用是终止某个case并跳出switch结构。当它出现在循环语句中，作用是跳出当前循环语句，执行后面的代码
* continue在循环语句中，立即终止本次循环，并执行下一次循环。
* goto语句可以跳出外层循环，需要对外层循环添加标记，然后使用goto语句可以跳转到标记的语句处。

2、for与while的相同之处都是完成一定次数的循环；其不同之处是：whie一般用于未知次数的循环；for一般用于次数已知的循环；

两者声明变量的作用域不同，如果定义一个变量i，在while循环后i的值会存在，但for循环后，i的值不存在了；

五、编程题

1、参考答案

#include<stdio.h>

void main()

{

int x = 5;

if (x % 2 == 0)

printf("%d为偶数\n",x);

else

printf("%d为奇数\n",x);

getchar();

}

2、参考答案

#include<stdio.h>

void main()

{

int i, sum = 0;

for (i = 0; i < 100; i++) //控制i在1-100之间

{

if (i % 2 != 0) //如果是奇数

sum += i; //就加起来

}

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

getchar();

}

# 第4章 函数

1. 填空题
2. return
3. 有参、无参
4. scanf()
5. 递归
6. 全局、局部
7. 判断题

1、√ 2、× 3、× 4、√ 5、√

1. 选择题

1、B 2、D 3、A、B、C 4、B、C 5、A、B、C

四、简答题

1、

* 局部变量

局部变量就是在函数内部声明的变量，它只在本函数内有效，也就是说，只能在本函数内使用它。此外，局部变量只有当它所在的函数被调用时才会被使用，而当函数调用结束时局部变量就会失去作用。

* 全局变量

全局变量(包括main()函数)，它不属于哪一个函数，而是属于源程序。因此全局变量可以为程序中的所有函数所共用。它的有效范围从定义处开始到源程序结束。

2、

* 外部函数

可以被其他源文件调用的函数称为外部函数，在调用外部函数之前，需要在当前源文件中定义外部函数。定义外部函数的方式是在函数的返回值类型前面添加extern关键字。

* 内部函数

函数只在它的定义文件中有效，该类函数称为内部函数。在定义内部函数时，需要在函数的返回值类型前面添加static关键字（又称为静态函数）。它不可以被别的源文件调用，只在当前源文件中有效。

五、编程题

1、参考答案

#include<stdio.h>

int \_sum(int i)

{

if (i == 1)

return 1;

if (i > 1)

return i + \_sum(i - 1); //函数自身调用自身

}

void main()

{

int sum;

sum = \_sum(10);

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

getchar();

}

2、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //scanf()函数是不安全的，调用它要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

int sum(int a,int b)

{

return a + b;

}

void main()

{

int a,b,he;

scanf("%d%d", &a,&b);//用户从键盘输入数据

he = sum(a, b);

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

system("pause"); //包含在stdlib.h头文件中，也可以让运行结果界面暂停

}

# 第5章 数组

1. 填空题
2. 集合
3. 首
4. 变量
5. 下标
6. 9
7. 判断题

1、√ 2、× 3、× 4、× 5、√

1. 选择题

1、D 2、A 3、C 4、A、C 5、A、B、C、D

四、简答题

1、一维数组指的是只有一个下标的数组，其定义方式如下所示：

类型说明符 数组名[常量表达式];

其初始化有三种方法：

* 直接对数组中的所有元素赋值；如int i[5]={1,2,3,4,5}；
* 只对数组中的一部分元素赋值；如int i[5]={1,2,3}；
* 对数组全部元素赋值，但不指定长度；如int i[]={1,2,3,4}；

2、一维数组和二维数组区别如下

* 一维数组只有一个下标，二维数组有两个下标；
* 二维数组有行与列之分，一维数组则没有；
* 赋值时一维数组下标可以略，二维数组可以省略第一个下标，第二个下标不能省略。

五、编程题

1、参考答案

#include<stdio.h>

#include<stdlib.h>

int \_max(int a[5]) //数组作为函数参数

{

int i;

int max = a[0];;

for (i = 1; i < 5; i++)

{

if (max < a[i])

max = a[i];

}

return max;

}

void main()

{

int a[5] = { 3, 4, 6, 9, 13 };

printf("max = %d\n", \_max(a));

system("pause");

}

2、参考答案

#include<stdio.h>

#include<stdlib.h>

void \_sort(int a[], int n)

{

int i, j, temp;

for (i = 0; i < n - 1; i++)

{

for (j = 0; j < n - i - 1; j++)

{

if (a[j] > a[j + 1])

{

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

}

void main()

{

int i;

int b[] = { 25, 24, 12, 76, 101, 96, 28 };

printf("排序前：");

for (i = 0; i < 7; i++)

printf("%d ", b[i]);

\_sort(b, 7); //数组名作为函数参数

printf("\n排序后：");

for (i = 0; i < 7; i++)

printf("%d ", b[i]);

system("pause");

}

# 第6章 指针

1. 填空题
2. 指针
3. 字符数
4. &
5. 函数指针
6. 二级指针
7. 判断题

1、√ 2、√ 3、× 4、√ 5、×

1. 选择题

1、C 2、A 3、B 4、C 5、A

四、简答题

1、在程序运行过程中，数据都是保存在内存中的，内存是以字节为单位的连续存储空间，每个字节都有一个编号，这个编号称为地址。变量也是有地址的，专门用于存放其他变量的地址，这种变量称为指针变量，通常简称指针。通过指针可以操作它指向的内存中的变量。取内存的地址或取内存中的值进行运算。如char arr[10],\*p;p = arr;arr是大小为10的指针数组，p是指向这个数组的指针。

2、数组指针是指向数组的指针，数组中存储的是统一的其他数据类型的变量；指针指向这一块内存的首地址。可以通过指针的移动来依次访问数组中的元素。

指针数组是指数组中存储的是指针，数组中指针的数据类型必须一致，指针数组的定义方式与普通数组类似，只需在类型前面加上“\*”，如int\*arr[10]; 存储了10个int类型的指针。

五、编程题

1、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //scanf()函数是不安全的，调用它要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

void swap(int \*a, int \*b)

{

int temp ;

temp = \*a;

\*a = \*b;

\*b = temp;

}

void main()

{

int a, b;

printf("请输入两个数：\n");

scanf("%d%d", &a, &b);

swap(&a, &b);

printf("交换后：\n%d %d\n", a, b);

system("pause");

}

2、参考答案

#include<stdio.h>

#include<stdlib.h>

void copystr(char \*form1, char \*form2)

{

while (\*form1)

{

\*form2 = \*form1;

form2++;

form1++;

}

\*form2 = '\0';

}

void main()

{

char str1[20] = "Hello World";

char str2[20] = "I am Chinese";

printf("复制前：\nstr1:%s\nstr2:%s\n", str1, str2);

copystr(str1, str2);

printf("复制后：\nstr1:%s\nstr2:%s\n", str1, str2);

system("pause");

}

# 第7章 字符串

1. 填空题
2. strlen()
3. 换行
4. b
5. strcmp()
6. 字符串
7. 判断题

1、× 2、× 3、√ 4、√ 5、×

1. 选择题

1、A 2、C 3、A、C、D 4、C、D 5、A

四、简答题

1、字符串比同等字符数组多占一个字符的内存空间。

2、Puts()函数接收的参数是一个字符串指针，该指针指向要输出的字符串，并且会自动在字符串末尾追加换行符‘\n’，调用成功时返回int类型的数，否则返回EOF。

strcat()函数在使用前，第一个参数对应的字符数组必须要有足够的空间来容纳连接后的字符串，否则会造成缓冲区溢出。

五、编程题

1、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //gets()也要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void main()

{

char s1[100], s2[100];

int i;

printf("请输入两个字符串：\n");

gets(s1);

gets(s2);

i = strcmp(s1, s2);

if (i > 0)

printf("第1个字符串较大\n");

else if (i < 0)

printf("第2个字符串较大\n");

else

printf("两个字符串相等\n");

system("pause");

}

1. 参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //gets()也要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void main()

{

int a, b, c;

char s1[100], s2[100],s3[100];

gets(s1);

gets(s2);

gets(s3);

a = atoi(s1);

b = atoi(s2);

c = atoi(s3);

printf("s1+s2+s3 = %d\n", a + b + c);

system("pause");

｝

# 第8章 预处理

1. 填空题
2. #define
3. #include
4. #if/#else/#endif、#ifdef、#ifndef
5. #include <文件名>、#include “文件名”
6. 880
7. 判断题

1、× 2、× 3、× 4、√ 5、√

1. 选择题

1、C 2、D 3、B 4、B 5、B

四、简答题

1、预处理指令有三种：宏定义、文件包含、条件编译。宏定义使用#define来表示，当一些常量在程序中频繁使用，为了避免书写错误，可以使用宏定义来表示。文件包含是指在C语言程序中经常会使用到一些函数，这些函数需要引用相关的头文件，引用方式为#include<stdio.h>或者是#include "stdio.h"。条件编译是指C语言程序为了完成某些功能，只有在指定条件下才进行编译。

2、条件编译的格式有三种。第一种是#if/#else/endif指令，只有符合判断条件才会执行相应的编译。第二种是#ifdef指令，如果想判断某个宏是否被定义可以使用该指令。 第三种是#ifndef指令，用来判断某个宏是否被定义，如果没有定义那么就会编译#ifndef到#endif中间的内容，否则就跳过。

五、编程题

1、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //gets()也要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

#define FUNC(x,y) x%y

void main()

{

int a, b;

scanf("%d%d", &a, &b);

printf("两个数的余数为：%d\n", FUNC(a, b));

system("pause");

}

1. 参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS //gets()也要关闭安全检查

#include<stdio.h>

#include<stdlib.h>

#define PI 3.1415926

#define CIR\_S(r) PI\*r\*r

#define CIR\_L(r) 2\*PI\*r

void main()

{

float r;

printf("请输入圆的半径：\n");

scanf("%f", &r);

printf("圆的面积：%.2f\n", CIR\_S(r));

printf("圆的周长：%.2f\n", CIR\_L(r));

system("pause");

}

# 第9章 结构体与共用体

1. 填空题
2. 构造
3. struct
4. union
5. stu.num
6. 8
7. 判断题

1、√ 2、√ 3、√ 4、× 5、×

1. 选择题

1、A 2、A C 3、C 4、B 5、 D

四、简答题

1、结构体类型的定义方式如下所示：

struct 结构体类型名称

{

数据类型 成员名1;

数据类型 成员名2;

…

数据类型 成员名n;;

};

共用体在定义上与结构体类型十分相似，共用体类型的定义方式如下所示：

union 共用体类型名称

{

数据类型 成员名1；

数据类型 成员名2；

……

数据类型 成员名n;

};

2、在程序执行的任何特定时刻，结构体变量中的所有成员是同时驻留在该结构体变量所占用的内存空间中，而共用体变量仅有一个成员驻留在共用体变量所占用的内存空间中。

五、编程题

1、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

//存储学生信息的结构体

struct ST

{

int ID;

char name[20];

float score[3];

};

//\_avg()求平均成绩的函数

float \_avg(struct ST \*s, int n)

{

int i, j;

int sum = 0;

float avg;

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

{

for (j = 0; j < 3; j++)

sum += s[i].score[j];

}

return (sum / n) / 3;

}

//\_get()函数找最高分学生，输出该学生的信息

void \_get(struct ST\* s, int n)

{

int i, j;

int max = 0; int index = 0;

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

{

if ((s[i].score[0] + s[i].score[1] + s[i].score[2]) > max)

{

max = s[i].score[0] + s[i].score[1] + s[i].score[2];

index = i;

}

}

printf("\n最高分学生信息：\n");

printf("学号：%d\n", (s + index)->ID);

printf("姓名：%s\n", (s + index)->name);

printf("成绩：%.2f,%.2f,%.2f\n", s[index].score[0], s[index].score[1], s[index].score[2]);

}

void main()

{

int i, j;

float avg;

struct ST sts[20];

//输入学生信息

for (i = 0; i < 20; i++)

{

printf("请输入第%d个学生信息：\n",i+1);

scanf("%d", &sts[i].ID);

scanf("%s", sts[i].name);

for (j = 0; j < 3; j++)

scanf("%f", &sts[i].score[j]); //输入各科成绩

}

printf("3门课的总平均成绩为：%.2f\n", \_avg(sts, 20));

\_get(sts, 20);

system("pause");

}

# 第10章 数据结构

1. 填空题

1、数据域 指针域

2、指针

3、栈顶 栈底

4、栈

5、入队 出队

1. 判断题

1、× 2、× 3、√ 4、× 5、√

1. 选择题
2. B 2、B 3、D 4、D 5、D
3. 简单题
4. 链表是一种常用的线性存储结构，它由一系列节点组成，节点可以在运行时动态生成，链表可以充分利用计算机内存空间，灵活实现内存动态管理。
5. 栈是只允许在其一端进行插入和删除操作的线性表，它遵循的是后进先出原则；而队列是一个线性表，元素是从表的一端进行插入，从表的另一端进行删除，它遵循的是先进先出的原则。
6. 编程题

1、参考答案

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

struct Node //结点

{

int data;

struct Node\* next;

};

struct Link //链表头

{

int length;

struct Node\* head;

};

void \_init(struct Link\* L) //初始化

{

L->length = 0;

L->head = NULL;

}

struct Link\* \_create()

{

int ret = 1;

struct Link\* pHead = (struct Link\*)malloc(sizeof(struct Link));

\_init(pHead); //初始化链表

struct Node\* pNode = NULL, \*temp = NULL;

printf("请输入数据，输入0结束\n");

while (ret)

{

scanf("%d", &ret);

if (ret)

{

pNode = (struct Node\*)malloc(sizeof(struct Node));

pNode->data = ret;

pNode->next = NULL;

if (temp == NULL)

temp = pNode;

else

temp->next = pNode;

if (pHead->head == NULL)

pHead->head = temp;

temp = pNode;

pHead->length++;

}

else

break;

}

return pHead;

}

//求链表的结点数

int \_getcount(struct Link\* pHead)

{

int count = 0;

struct Node\* temp = pHead->head;

if (pHead == NULL)

return -1;

while (temp)

{

count++;

temp = temp->next;

}

return count;

}

int main()

{

int num;

struct Link\* pHead = NULL;

//创建链表

pHead = \_create();

printf("此链表有%d个结点\n", \_getcount(pHead));

system("pause");

return 0;

}

# 第11章 文件操作

1. 填空题

1、文本文件 二进制文件

2、ASCII码

3、文件有关信息

4、流

5、fopen() fclose()

二、判断题

1、× 2、× 3、√ 4、√ 5、×

三、选择题

1、B 2、A 3、C 4、C 5、ABC

四、简答题

1. 在C语言中，通过不同输入/输出设备之间的数据传输抽象表述为“流”，流实际上是一个字节序列，输入程序的字节序列被称为输入流，从程序输出的字节序列被称为输出流。
2. 文件指针是一个指向文件有关信息的指针，这些信息包括文件名、状态和当前位置，它们保存在一个结构体变量中；文件位置指针是随机读写文件的关键，它可以实现对文件任何位置进行读取的操作。
3. C语言提供了一个fopen()函数的，该函数用于打开文件，其返回值类型为文件指针。而fclose()函数用于关闭文件，其返回值为int类型

五、编程题

1、参考答案

#include<stdio.h>

#include<stdlib.h>

int main()

{

char buf[1024];

FILE\* fp1,\*fp2;

fp1 = fopen("C:\\Users\\www\\Desktop\\a.txt", "r");

if (fp1 == NULL)

printf("文件a.txt打开失败\n");

fp2 = fopen("C:\\Users\\www\\Desktop\\b.txt", "w");

if (fp2 == NULL)

printf("文件b.txt打开失败\n");

while (!feof(fp1))

{

fgets(buf, 1024, fp1);

fputs(buf, fp2);

}

printf("拷贝完成\n");

fclose(fp1);

fclose(fp2);

system("pause");

return 0;

}