

C语言刷题训练营-第一讲

比特IT读书会@提供课程讲解和技术支持

1. BC1-实践出真知

于老师经常告诉我们“学习编程最好的办法就是上机实践，因为你要对计算机下指令，想让计算机帮你干活，就得多和计算机‘交流’，实践才能出真知。”

输入描述:

无

输出描述:

Practice makes perfect!

参考代码:

```
#include <stdio.h>

int main()
{
    printf("Practice makes perfect!");
    return 0;
}
```

解析:

本题是一个没有输入要求的题目，只考察输入，所以只需要准确无误的输出结果就行了。

2. BC2-我是大V

题目描述

每个人都想成为大V (VIP: Very Important Person)，但要一点一点积累才行，先从小v做起。要求输出由小写字母v组成的大V。

输入描述:

无

输出描述:

```
v    v
v  v
v
```

备注:

换行使用转义字符'\n'

参考代码:

```
#include <stdio.h>
//代码1
int main()
{
    printf("v    v\n");
    printf("v  v\n");
    printf("v\n");
    return 0;
}

//代码2
#include <stdio.h>
int main()
{
    //一行输出的效果
    printf("v    v\n v  v\n v\n");
    return 0;
}
```

答案解析:

这个题也是**没有输入要求**的题目，所以相对比较简单。

题目中只要能够找到输出的规律和掌握'\n'的使用就没问题。

这里简单提及：转义字符，可以做一个拓展。

3. BC3-有容乃大

题目描述

确定不同整型数据类型在内存中占多大（字节），输出不同整型数据类型在内存中占多大（字节）。

输入描述:

无

输出描述:

不同整型数据类型在内存中占多大（字节），具体格式详见输出样例，输出样例中的?为不同整型数据类型在内存中占的字节数。输出样例如下：

The size of short is ? bytes.

The size of int is ? bytes.

The size of long is ? bytes.

The size of long long is ? bytes.

参考代码

```
#include <stdio.h>

int main()
{
    printf("The size of short is %d bytes.\n", sizeof(short));
    printf("The size of int is %d bytes.\n", sizeof(int));
    printf("The size of long is %d bytes.\n", sizeof(long));
    printf("The size of long long is %d bytes.\n", sizeof(long long));

    return 0;
}
```

答案解析：

这类入门的题目基本都是没有输入操作的。

本题的关键是要掌握C语言的 sizeof 这个操作符。

sizeof 是C语言的一个单目操作符，用来计算不同类型数据所占内存空间的大小，单位是字节。

4. BC7-缩短2进制

题目描述

我们处理的整数通常用十进制表示，在计算机内存中是以二进制补码形式存储，但通常二进制表示的整数比较长，为了便于在程序设计过程中理解和处理数据，通常采用八进制和十六进制，缩短了二进制补码表示的整数，但保持了二进制的表达特点。请输出十进制整数1234对应的八进制和十六进制。

输入描述:

无

输出描述:

十进制整数1234对应的八进制和十六进制（字母大写），用空格分开，并且要求，在八进制前显示前导0，在十六进制数前显示前导0x。

备注:

printf可以使用使用格式控制串“%o”、“%x”分别输出八进制整数和十六进制整数，并使用修饰符“#”控制前导显示

参考代码：

```
#include <stdio.h>
int main()
{
    printf("0%o 0x%x", 1234, 1234);
    return 0;
}
```

答案解析：

这里需要我们掌握对各种数据的格式控制

比如：

%c - 字符
%hd - 短整形
%d - 整形
%s - 字符串
%f - 单精度浮点数
%lf - 双精度浮点数
%p - 地址格式
...还有很多

格式参考：<https://zh.cppreference.com/w/c/io/fscanf>

注意：注意还是严格按照题目要求的格式输出，才能保证正确性。

5. BC19-反向输出4位数

题目描述

将一个四位数，反向输出。

输入描述：

一行，输入一个整数 n ($1000 \leq n \leq 9999$)。

输出描述：

针对每组输入，反向输出对应四位数。

示例1

输入

1234

输出

4321

参考代码：

```
#include <stdio.h>
int main()
{
    int n = 0;
    //输入数据
    scanf("%d", &n);
    //获取每一位
    while(n)
    {
        printf("%d", n%10);
        n /=10;
    }
    return 0;
}
```

答案解析:

这里得掌握操作符%和/的基本使用和特点。其次是循环的使用。

6. BC28-大小写转换

题目描述

实现字母的大小写转换。多组输入输出。

输入描述:

多组输入，每一行输入大写字母。

输出描述:

针对每组输入输出对应的小写字母。

示例1

输入

A
B

输出

a
b

备注:

多组输入过程中要注意“回车”也是字母，所以要“吸收”(getchar())掉该字母。

参考答案:

```
//代码1-这个代码对吗?  
#include <stdio.h>  
  
int main()  
{  
    int ch = 0;  
    //循环数据输入  
    while((ch=getchar()) != EOF)  
    {  
        printf("%c\n", (ch+32));  
    }  
    return 0;  
}
```

```
//代码2-正确的代码实现  
#include <stdio.h>  
  
int main()  
{  
    int ch = 0;  
    //循环数据输入  
    while((ch=getchar()) != EOF)  
    {  
        printf("%c\n", (ch+32));  
        getchar(); //处理多余的\n字符  
    }  
    return 0;  
}
```

答案解析：

1. 多组数据输入的问题。
2. `getchar` 清理缓冲区的问题，这里得理解 输入缓冲区 的问题。

视频中讲解。

比特IT读书会@提供课程讲解和技术支持

联系鹏哥：15596668862（同微信）