实验二　基本数据类型的操作（一）

一、实验目的

（1）掌握一个 C 语言源程序的完整结构。

（2）掌握 C 语言的数据类型，熟悉如何定义一个整型、字符型和浮点型的变量以及对它们赋值的方法。

（3）能对变量正确赋值，掌握不同类型数据之间赋值的规律。

（4）学会使用 C 的有关算术运算符，以及包含这些运算符的表达式，特别是自加（++）和自减（--）运算符的使用。

（5）掌握对运算符和表达式的正确使用。

（6）进一步熟悉 C 程序的编辑、编译、连接和运行过程。

（7）熟悉对程序的调试过程。

二、预习知识

（1）一个 C 语言源程序的结构。

（2）各种不同类型变量的定义方式。

（3）赋值的相关知识。

（4）各种运算的优先级和结合方式。

（5）有关表达式的相关知识。

三、实验内容

（1）输入并运行下面的程序。

```
main()
{
  char c1,c2;
  c1='a';
  c2='b';
  printf("%c%c\n",c1,c2);
}
```

1）运行此程序。

2）在此基础上增加一个语句：

```
printf("%d %d\n",c1,c2);
```

运行并分析结果。

3）将第 2 行改为：

```
int c1,c2;
```

运行并观察结果。

4）将第 3、4 行改为：

```
c1=a;     /*不用单撇号*/
c2=b;
```

运行并分析其运行结果。

5）将第 3、4 行改为：

```
c1="a";    /*用双撇号*/
c2="b";
```

运行并分析其运行结果。

6）再将第 3、4 行改为：

```
c1=300;         /*用大于 255 的整数*/
c2=400;
```

运行并分析其运行结果。

（2）写出下列程序，按照要求对输出结果进行分析，比较分析结果和显示结果，体会数据表示的规律。

```
main()
{
   int x=010,y=10,z=0x10;
   printf("x=%o,y=%d,z=%x\n",x,y,z);
}
```

1）分析程序的输出结果，体会" %o"，"%d"， "%x "数据类型输出的结果。

2）将程序的第 4 行改为：

```
printf("x=%d,y=%d,z=%d\n",x,y,z);
```

分析输出结果与显示的输出结果，体会十进制、八进制和十六进制数据之间的转换规律。

3）将程序的第 3 行改为：

```
char c1='M',c2='\x4d',c3='\115',c4=77,c;
```

第 4 行改为：

```
printf("c1=%c,c2=%x,c3=%o,c4=%d\n",c1,c2,c3,c4);
```

分析输出结果是什么，再调试运行后看显示结果，比较、分析得出其中的规律。（注意看调试后的显示结果，是否有前导 0 或 0x）。

4）将程序的第 4 行改为：

```
printf("c1=%c,c2=%c,c3=%c,c4=%c\n",c1,c2,c3,c4);
```

分析输出的结果是什么，再调试运行后看显示结果，通过分析比较，你能发现什么？（结果都是什么？字符可以自由的表示成几种形式？）

5）将程序的第 4 行改为：

```
printf("c1=%d,c2=%d,c3=%d,c4=%d\n",c1,c2,c3,c4);
```

分析输出的结果是什么，运行调试后看显示结果，通过分析比较，你能发现什么？

6）将程序的第 4 行更换成以下两行：

```
c=c1+32;
printf("c=%c,c=%d\n",c,c);
```

分析输出的结果是什么，运行调试后看显示结果，通过分析比较，你能发现什么？

（3）输入以下程序 。

```
main()
{
  int i,j,m,n;
```

```
  i=8;
  j=10;
  m=++i;
  n=j++;
  printf("%d,%d,%d,%d",I,j,m,n);
}
```

1）运行程序，注意 i、j、m、n 各变量的值，分别做以下改动并运行。

2）将第 4、5 行改为：

```
m=i++;
n=++j;
```

再运行。

3）将程序改为：

```
main( )
{
  int i,j;
  i=8;
  j=10;
  printf("%d,%d",i++,j++);
}
```

4）在 3）的基础上，将 printf 语句改为：

```
printf("%d,%d",++i,++j);
```

5）再将 printf 语句改为：

```
printf("%d,%d,%d,%d",i,j,i++,j++);
```

6）将程序改为

```
main()
{
  int i,m=0,n=0;
  i=8;
  j=10;
  m += i++;n -= --j;
  printf("i=%d,j=%d,m=%d,n=%d",i,j,m,n);
}
```

四、实验小结

（1）C 语言中整型数据、一般的字符型数据分别可以以什么形式表示？

答：整型数据可以用十进制、八进制、十六进制形式表示。

一般字符型数据可以用字符本身（例'M'）或转义表示。转义表示可以用字符的八进制或十六进制 ASCII 码值表示（如'\115'、'\x4d'）。

（2）程序（2）中 c4=77,c=c4+32 说明什么问题？

答：C 语言中字符型数据存储的是字符的 ASCII 码值（1 个字节的整数），所以可以将一个整数（看作字符的 ASCII 码）赋值给字符变量。本实验即允许 c4=77。同时，C 语言字符数据可以参与算术运算，其本质是字符数据的 ASCII 码值（整数）参与算术运算。本实验即允许 c4+32，其结果仍然是一个整数。再将结果看作是一个字符的 ASCII 码值，赋值给字符变量 c，即本实验中的 c=c4+32。

另外，从（2）可以看出同一字母小写大写相差 32。即大写字母 ASCII=小写字母的 ASCII+32。

（3）程序（1）中 c=77 是否还可以写成 c=0x4d，c=0115？

答：可以，整数可以直接赋值给字符变量，即十进制、八进制、十六进制形式的整数都可以赋值给字符变量。