

HW1: 数据的表示

实验目的：熟悉数值数据在计算机内部的表示方式，掌握相关的处理语句。

实验报告要求：

1. 说明你做实验的过程（重要步骤用屏幕截图表示）。
2. 提交你用于解题的源程序（可以直接附在报告中）。
3. 分析或回答问题。

完成下列实验，提交实验报告：

1. 下述两个结构体所占存储空间多大？结构中各分量所在位置相对于结构体起始位置的偏移量是多少？请编写程序以验证你的答案。若使用 `#pragma pack(2)` 语句，则结果又如何？若使用 `#pragma pack(32)` 语句，结果又如何？

```
1 struct test1{
2     char x1[15];
3     short x2[3];
4     int x3;
5     long long x4;
6 };
7
8 struct test2
9 {
10     char x1[15];
11     short x2[3];
12     int x3;
13     long long x4;
14 }__attribute__((aligned(16)));
```

2. `-1 < 1` 和 `-1 < 1u` 的结果一样吗？为什么？

3. 运行以下程序代码，并对程序输出结果进行分析。

```
1 #include <stdio.h>
2
3 void main(){
4     unsigned int a = 1;
5     unsigned short b = 1;
6     char c = -1;
7     int d;
8
9     d = (a > c) ? 1 : 0;
10    printf("unsigned int is %d\n", d);
11    d = (b > c) ? 1 : 0;
12    printf("unsigned short is %d\n", d);
13 }
```

4. 运行下列代码，并对输出结果进行分析。

```
1  #include <stdio.h>
2
3  void main(){
4      union NUM{
5          int a;
6          char b[4];
7      } num;
8      num.a = 0x87654321;
9      printf("0x%X\n", num.b[1]);
10     printf("0x%X\n", num.b[3]);
11 }
```

5. 请说明下列赋值语句执行后，各个变量对应的机器数和真值各是多少？编写一段程序代码并进行编译，观察默认情况下，编译器是否报warning。如果有warning信息的话，分析为何会出现这种warning信息。

```
1  int a = 2147483648;
2  int b = -2147483648;
3  int c = 2147483649;
4  unsigned short d = 65539;
5  short e = -32790;
```

6. 完成书上第二章习题中第40题，提交代码，并在程序中以十六进制形式打印变量 `u` 的机器数。

第一版第二次印刷的课本上第40题是函数 `fpower2` 的程序填空，若新版本课后习题序号不同，请完成 `fpower2` 程序填空所对应的题目。

7. 编译运行以下程序，并至少重复运行3次。

```
1  #include <stdio.h>
2
3  void main(){
4      double x = 1.001, y = 0.001, z = 1.0;
5      for (int i = 0; i < 10; i++) {
6          if ((x-y) == z)
7              printf("equal\n");
8          else
9              printf("not equal\n");
10         x += z;
11         y += z;
12         printf("#%d: %f, %f\n", i, x, y);
13     }
14 }
```

要求：

- 1) 给出每次运行的结果截图。
- 2) 每次运行过程中，是否每一次循环中的判等结果都一致？为什么？
- 3) 每次运行过程中，每一次循环输出的 `i`、`x` 和 `y` 的结果分别是什么？为什么？

8. 考虑以下C代码：

```
1 int x = -1;
2 unsigned u = 2147483648;
3 printf("x = %u = %d\n", x, x);
4 printf("u = %u = %d\n", u, u);
```

在32位机器上运行上述代码时，它的输出结果是什么？为什么？

9. 回答以下问题：

- 1) 在有些32位系统上，C表达式 `-2147483648 < 2147483647` 的执行结果为 `false`。Why?
- 2) 若定义变量 `int i = -2147483648;`，则 `i < 2147483647` 的执行结果为 `true`。Why?
- 3) 如果将表达式写成 `-2147483647-1 < 2147483647`，则结果会怎样呢？Why?

10. 考虑以下C代码：

```
1 #include <stdio.h>
2
3 void main(){
4     int x = -1;
5     unsigned u = 2147483648;
6     printf("x = %u = %d\n", x, x);
7     printf("u = %u = %d\n", u, u);
8
9     if(-2147483648 < 2147483647){
10         printf("-2147483648 < 2147483647 is true\n");
11     } else {
12         printf("-2147483648 < 2147483647 is false\n");
13     }
14
15     if(-2147483648-1 < 2147483647){
16         printf("-2147483648-1 < 2147483647\n");
17     } else if(-2147483648-1 == 2147483647){
18         printf("-2147483648-1 == 2147483647\n");
19     } else{
20         printf("-2147483648-1 > 2147483647\n");
21     }
22 }
```

在C90和C99标准下运行的结果相同吗？为什么？