

cs-easy-01

step 1

```
#include <stdio.h>
#include <string.h>
int main()
{
    printf("My name is likaishuo.\n");
    printf("I am 18 years old.\n");
    return 0;
}
```

问题1: #include是一个预处理指令, <stdio.h>是一个头文件, 头文件中包含printf这个函数, 可以使我能够在下面的代码中调用这个函数

问题2: main是主函数, 即代码的主体, 而main函数的返回值为整数 (return 0), 因此要使用int来定义main。

再询问ai后, 我了解到, c语言规定了main函数作为程序的入口点

问题3: int用来存储整型, 即整数类型的量, 如数组, 变量, 指针等

基本数据类型有整型, 字符, 字符串, 浮点数

问题4: \n是换行符, 让两个printf函数输出的内容在两行中分别出现, 不会连在一起。而通过询问ai, 让我想起, windows当中, \n并非换行功能, \n的实质是\r\n, 由编译器帮我完成这一个转换

enter的作用是换行和确认, 通过询问ai, 我了解到, 回车是两个动作, 回到行首和往下一行, 即\r和\n。
\

step2

问题1：变量的类型即数据类型，不同的数据类型决定了汇编语言的翻译结果

如整数类型若用char来输入，会获得一个对应的字母或符号。同时数据类型还与内存分配有关，不同的变量类型对应的内存长度不同。

选择数据类型应先看数据是什么类型的（整数、浮点、字符或字符串），再看数据长度，翻译为二进制后是否过长（选择int或long），如25用int即可。而字符串比较特殊，字符串的本质是一个由字符组成的数组，不能直接通过char类型的字符变量实现，需要用char*定义字符串的数组指针，或者用char a[999] = "apple"来定义

问题2：数组第一个元素的下标为0，即arr[0]；同时第一个元素的指针与这个数组的指针相同

访问时，会得到一个与数组无关的数字，读取位置的内存内容，带来程序的不确定性。

而输入过程如果越界，可能会改变非数组内存中数据的值，破坏程序，因此数组不能越界，字符数组[]中的数字可以适当写大

问题3：for循环在括号里输入i（初始化），i的条件（条件判断），以及i自增（迭代），如果i满足条件代码就能一直运行，实现循环

而while循环在函数外定义变量i，在括号内输入i的条件，在循环里定义i的自增，在相同的循环逻辑下实现循环

初始化的作用是定义i的值，若无初始化i为一个任意值，循环会出错

条件判断的作用是判断i的值经过变化后是否符合条件，控制循环的结束

迭代用来更新变量，使循环过程中i的值一直在改变，若不迭代，循环不会停止

while函数先判断条件再进行循环，do while函数先进行循环在判断条件，相同的条还能控制下，do while函数会多运行一次

```
#include <stdio.h>

int main()
{
    int sum = 0;
    int i = 1;
```

```

while(i <= 10)
{
    sum = sum + i;
    i ++;
}
printf("%d",sum);
return 0;

}

```

```

#include <stdio.h>
int main()
{
    int sum = 0;
    int i = 1;
    do
    {
        sum = sum + i;
        i ++;
    }while(i < 10);
    printf("%d",sum);
    return 0;

}

```

问题4:

	算数表达式	逻辑表达式
运算对象	数值（数的运算）	布尔值
运算结果	一个数	（是非0数；否0）

&&表示并且，要求前后两个条件同时成立；||表示或者，要求前一个成立或者后一个成立，有一个成立即可

！表示取反，用于逻辑计算，非零输出0,0输出非零


```

    }
}

a:return 0;
}
//另外, {与函数写在一行, 我认为这样会降低代码的阅读性, 因此在我的完整代码中, 我修改了相应的格式

```

如上图是我的修改, 下展示完整代码

```

#include <stdio.h>

int main()
{
    int age;
    char name[20];

    printf("请输入你的姓名: ");
    scanf("%s", &name);
    printf("请输入你的年龄: ");
    scanf("%d", &age);
    while(1)
    {

        int flag = 0, count = 0;
        b: printf("My name is %s.\n", name);
        printf("I am %d years old.\n", age);
        printf("是否继续输入? (若继续请输入1, 反之请输入0): ");
        scanf("%d", &flag);
        count ++;
        if(flag == 1)
        {
            printf("请输入你的姓名: ");
            scanf("%s", &name);
            printf("请输入你的年龄: ");
            scanf("%d", &age);
            goto b;
        }
        else{

```

```

        printf("一共输入了%d次", count);
        goto a;
    }
}

a: return 0;
}

```

step3

这个板块对我理解函数起了很大的帮助，在之前看网课的学习中，我没有思考自定义函数的运行逻辑，以及其逻辑组织的特点

以下是我对“趣味赛代码”的改写

```

#include <stdio.h>
#include <string.h>
void paixu(int fenshu[3], char name[3][100])
{
    for(int i = 0; i < 2; i++)
    {
        for(int j = 0; j < 2 - i; j++)
        {
            if(fenshu[j] < fenshu[j + 1])
            {
                int temp = fenshu[j];
                fenshu[j] = fenshu[j + 1];
                fenshu[j + 1] = temp;
                char tname[100];
                strcpy(tname, name[j]);
                strcpy(name[j], name[j + 1]);
                strcpy(name[j + 1], tname);
            }
        }
    }
}

int pingjunshu(int a, int b, int c)

```

```

{
    int sum = (a + b + c)/3;
    int fang = ((sum - a) * (sum - a) + (sum - b) * (sum - b) + (sum - c) * (sum - c)) /
3;
    int pingjunshu = 3 * sum - fang / 3;
    return pingjunshu;
}

int main()
{
    int x1, x2, x3;
    int y1, y2, y3;
    int z1, z2, z3;

    printf("请输入小明的三项成绩（顺序为A B C,以一个空格为间隔）：");
    scanf("%d %d %d", &x1, &x2, &x3);
    printf("请输入小强的三项成绩（顺序为A B C,以一个空格为间隔）：");
    scanf("%d %d %d", &y1, &y2, &y3);
    printf("请输入小林的三项成绩（顺序为A B C,以一个空格为间隔）：");
    scanf("%d %d %d", &z1, &z2, &z3);
    int zh1 = pingjunshu(x1, x2, x3);
    int zh2 = pingjunshu(y1, y2, y3);
    int zh3 = pingjunshu(z1, z2, z3);
    //为了进一步简化函数，可以使用冒泡排序，利用数组来进行比较
    //但是，在使用冒泡排序的过程中，我只能比较出三个平均数得大小，而不能替换为三个人名
    //因此， 可以使用双数组来优化这个代码
    int fenshu[] = {zh1, zh2, zh3};
    char name[][100] = {"小明", "小强", "小林"};
    paixu(fenshu,name);

    printf("三人平均成绩从高到低的顺序为： %s %d, %s %d, %s %d", name[0], fenshu[0],
name[1], fenshu[1], name[2], fenshu[2]);

    return 0;
}

```

值得一提的是，在这个代码的书写过程中，我忘记了冒泡排序的正确书写方法，在初次书写时使用了单循环，导致了单方向的比较；而在自己的改动中，搞错了比较的量，第一次更改时，将内循环j与j + 1的比较写成了i与j的比较，将冒泡排序与选择排序混淆

通过这个练习，我复习了数组排序的知识

问题2：该函数进行了值传递，在swap函数中只改变了传递值，a，b指针所对应的值没有改变，因此printf出的值还是a = 10;b = 20;

以下关于指针问题的学习与解答在第二题中，以上是我对第一题的全部作答

我将把以上涉及的函数另存在.c中上传

由于我的语言组织能力一般，不能像有的人在回答中穿入很多诙谐的语言，使得我的回答人机感很重。但我保证，我的作答切实反映了我思考的过程，并非ai生成。这样的作答可能会让阅卷学长感到沉闷，还望见谅