

CS-EASY-01 What's your name

文字部分回答

from 2025090907001 赵子睿

PART 1:

1. 解释一下#include <stdio.h>的含义，分析在上述程序中它的作用。
 - 答：#include是预处理命令，<stdio.h>是头文件，其中包含了一些函数如输出函数printf(),#include<stdio.h>含义为预先找到<stdio.h>并将其包含到当前程序中。在上述程序中起到了将<stdio.h>中的输出函数printf()预先包含进程序中，以便后面程序调用该函数进行年龄的输出。
2. main函数在整个程序中起到什么作用？为什么main函数要定义为int类型
 - 答：main 函数是程序的主入口，且main 函数中的一些函数可以接收命令行输入并向操作系统传递程序的执行结果状态。因为一般来说返回值为 0 表示程序正常结束，返回非 0 值表示程序异常终止，返回值都是整数，所以main前面的数据类型必须与返回值对应为整数，即用int来定义。
3. int类型是计算机存储什么元素的方式？了解一下C语言中的基本数据类型。
 - 答：int 类型是用于存储整数的方式，C 语言的基本数据类型主要有整数类型、浮点类型和字符类型三种。
4. 转义字符\n的含义是什么？了解一下常见的转义字符及其含义，回车键本质上是什么。
 - 答：\n表示换行符，使光标换到下一行。回车键本质是\r\n,会同时触发 \n和 \r的效果，即光标移到到下一行并回到开头。

PART 2:

1. 变量类型:
 - i. 数据类型主要有整数类型、浮点类型和字符类型三种。
 - ii. 存放年龄应用整型。
 - iii. 不能实现，应在变量名后加上[x],x为变量中内容所占字节数。
2. 数组的起始与边界:
 - i. 从0开始。

ii. 导致访问数字不为数组内数字

iii. 常见在于数组访问常与循环结合，若循环条件写错，很容易越界且编译器不会报错；危险在于可能会出现安全漏洞，数据损坏与程序异常等。

3. 流程控制 - 循环结构：

i. 控制模式：最开始进行初始化，接着进行条件判断，若成立，**则执行一次循环体并进行一次迭代，然后再进行条件判断**，若成立则重复上方加粗操作，若不成立则循环结束。

ii. for与while的格式：

for格式：

```
for (初始化; 条件判断; 迭代) {  
    循环体  
}
```

while格式：

初始化

```
while (条件判断) {
```

循环体

迭代

```
}。
```

iii. do...while循环会先执行一次循环体与迭代再进行条件判断，而while循环则先进行判断再执行循环体与迭代。

4. 流程控制 - 逻辑表达式：

i. 算数表达式用于数值计算，生成具体的数值结果，逻辑表达式核心是得出真或假的结论，为条件判断提供依据。

ii. &&是逻辑与，其含义为当左右两侧的条件同时为真时，整体结果为真，否则为假。|是逻辑或，其含义为当左右两侧的条件至少有一个为真时，整体结果为真；否则为假。!是逻辑非，其含义为反转条件的真假。

iii. 举例：

```
int score = 75;  
int result = !(score < 60) && (score <= 100);
```

预测结果为真，即result = 1.

5. 将小明程序中的错误点一一列出，陈述你认为它出错的原因，并更正：

i. 输入输出的语句应在循环中

ii. 输入语句的name前面没有&

iii. 将变量count赋值为0应在循环前面而不是在循环内(与此同时定义也要在循环外，赋值前)

iv. while后括号中的条件判断语句ture指代不明，可直接用1代替用来无限循环

v. count++应在循环内的输入输出语句之后，而不是在是否继续执行的判断语句内，在判断语句内会导致最终获取count偏小

vi. printf("是否继续输入？（若继续请输入1，反之请输入0）："); 语句中的分号是中文符号，应该为英文符号的分号

vii. 接下来展示我更正后的代码：

```
int main() {
int age;
char name[20];
int count = 0;
int flag = 0;

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

    printf("My name is %s.\n", name);
    printf("I am %d years old.\n", age);
    count++;
    printf("是否继续输入? (若继续请输入1, 反之请输入0): ");
    scanf("%d", &flag); // 假设此处输入只会为0或1

    if (flag == 1) {
        continue;
    }
    else {
        printf("一共输入了%d次", count);
        break;
    }
}

return 0;
}
```

PART3

尝试从值传递的角度入手, 说说为什么会这样:

- 答: 该代码采用的是值传递, 这就导致实参a, b仅仅只是复制自身的数值并传递到形参中, 所以在 swap () 函数进行交换时实际上只交换了形参中存储的数值, 当swap () 函数运行结束后, 形参中的数值就被释放掉了, 不会影响主函数main () 中实参a,b中的数值。

下面是我修改part2中代码的运行结果以及part3中改进的代码的运行结果：

[点此处查看结果图片1]:<https://imgse.com/i/pVRsS9H>

[点此处查看结果图片2]:<https://imgse.com/i/pVRrx4e>

至此工序已毕，只待前辈检阅，如有不足之处，还望指正，谢谢诸位 😊😊😊