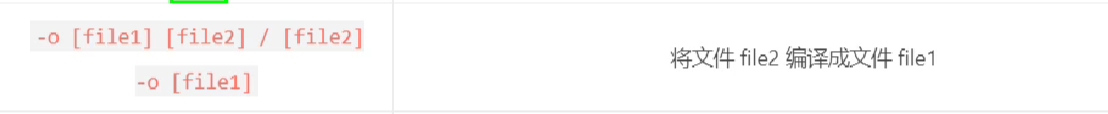


（g++ -E main.cpp -o main.i

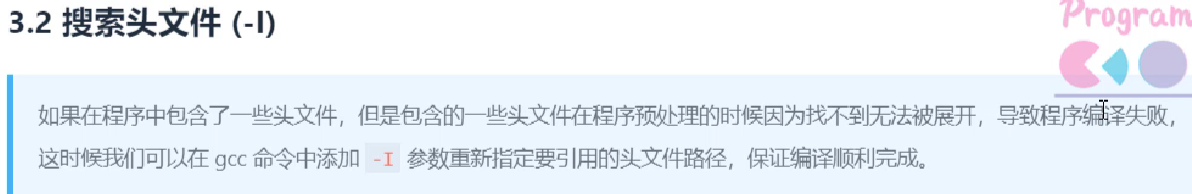
g++ -S main.cpp -o main.s

g++ -C main.cpp -o main.o

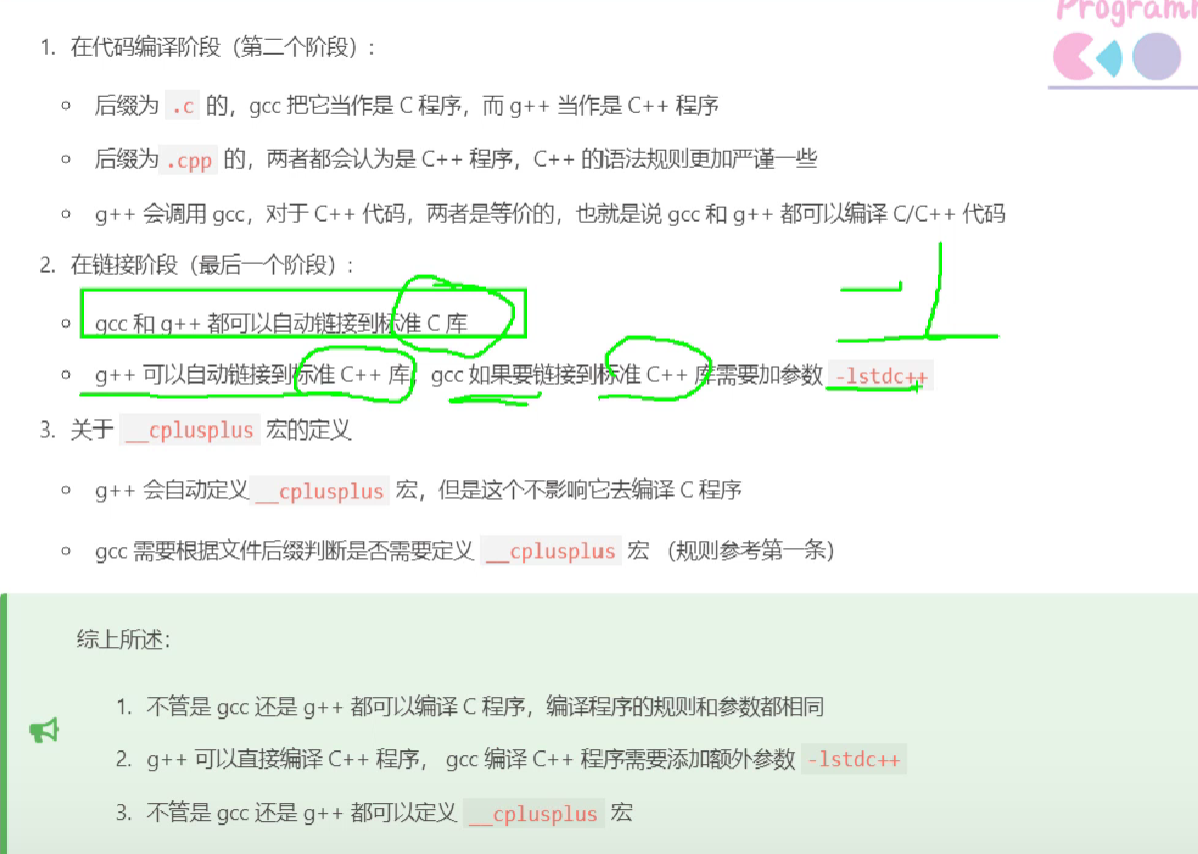
g++ main.o -o main ） ===》 g++ -o main main.cpp/ g++ main.cpp -o main







g++ -o test test.cpp -D DEBUG (-D 向文件中导入一个宏)



gcc -o main main.cpp -lstdc++





gcc args.c -g -Wall -O0 -o app

gcc -o app args.c -g -Wall -O0

gdb app(可调试的执行文件)

l （查看 调试的可执行文件的）

//向调试的可执行文件 传入参数

Set args (传入的参数)

//展示传入的参数

Show args

start (只会执行main中的第一行)

Run (可以一直执行下去，遇到断点会停下来)

q（离开调试）

list（l） 可以查看调试的文件中的指定信息



List

List n(数字)

List main(函数名)



L main.cpp: 2

L mian.cpp : main



break == b

b 行号

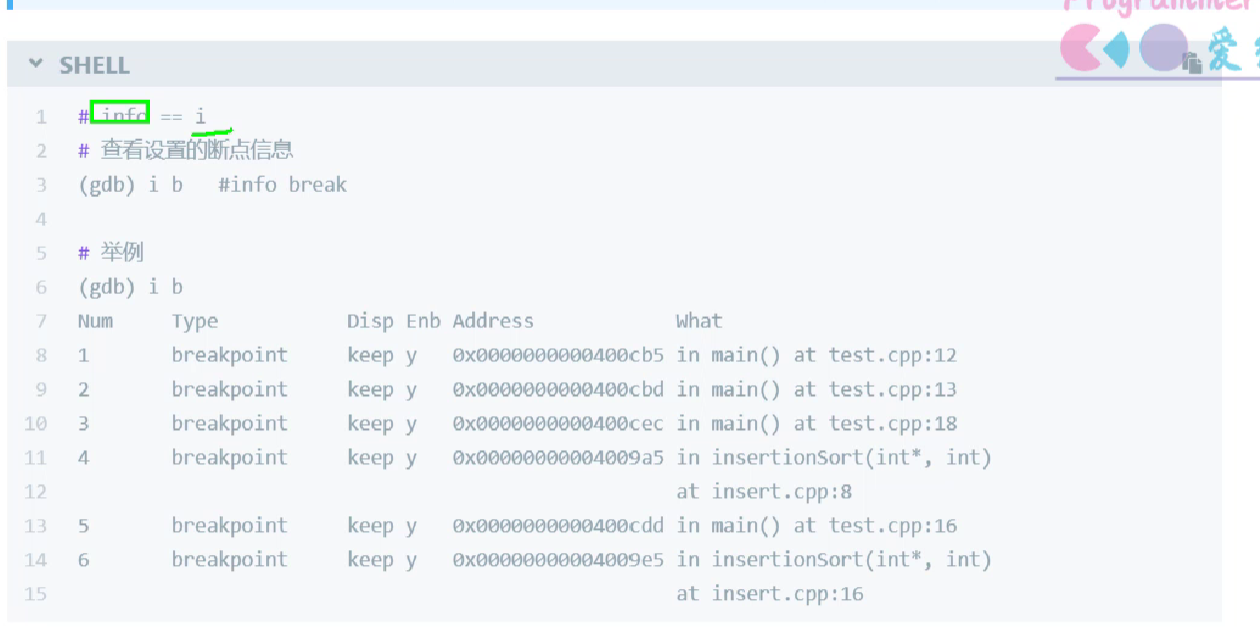
b 函数名 #停止在函数的第一层

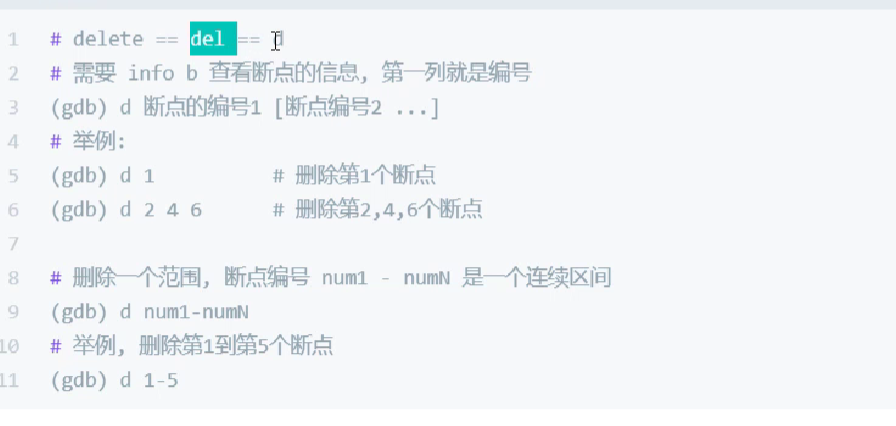
b 文件名 : 行号

b 文件名 ：函数名 # 停止在有函数的第一行



b 行数 if 变量名 == 某个值





d n (断点的编号)

d n1-n2 (是一个连续的区间)



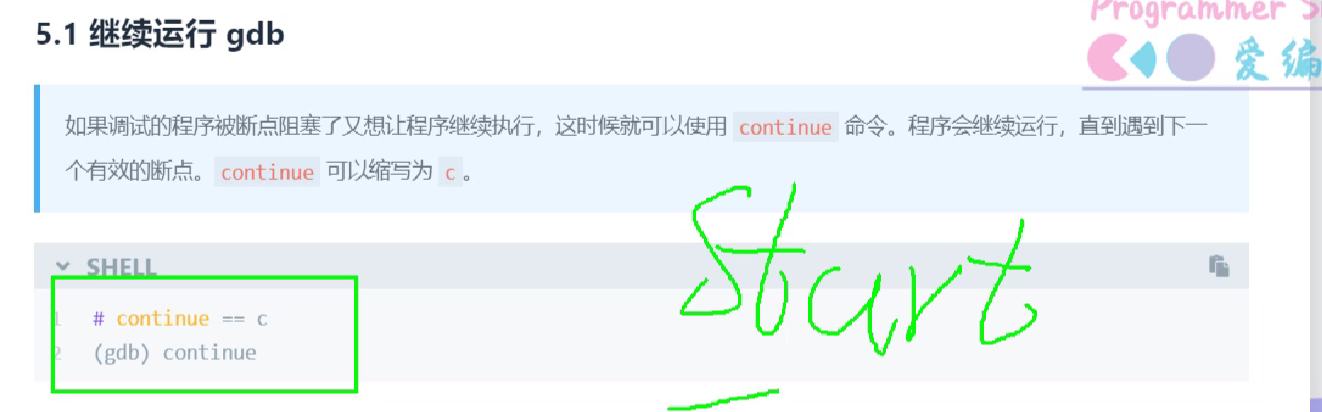
dis n （断点的编号）

dis n2-n1 (断点的区间)



ena n (断点1的编号)

ena n2-n3(断点的区间)



Continue -- 与 start 配对

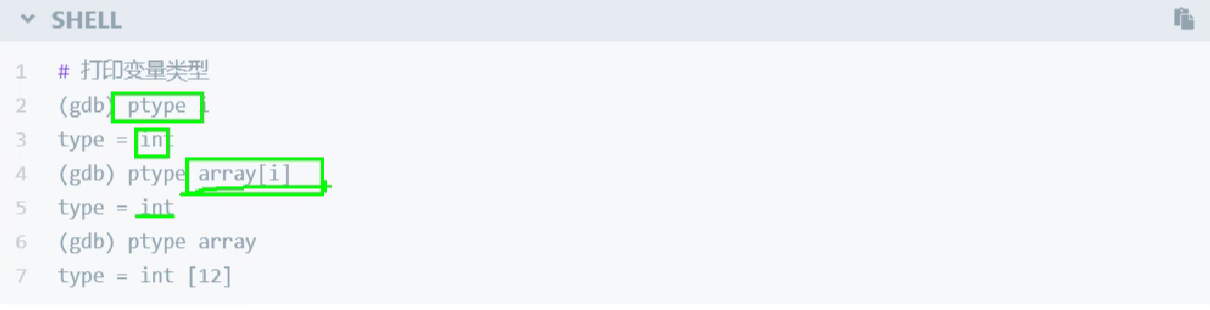


  
p 变量名

p i # 10进制

p / x i # 16进制

p /o i # 8 进制



ptype 变量名



display 变量名

自动显示追踪的变量名



undisplay num

undisplay num1-num2



重新启动变量的追踪：

enable display num --ena

取消对应变量的追踪:disable display num1-num2

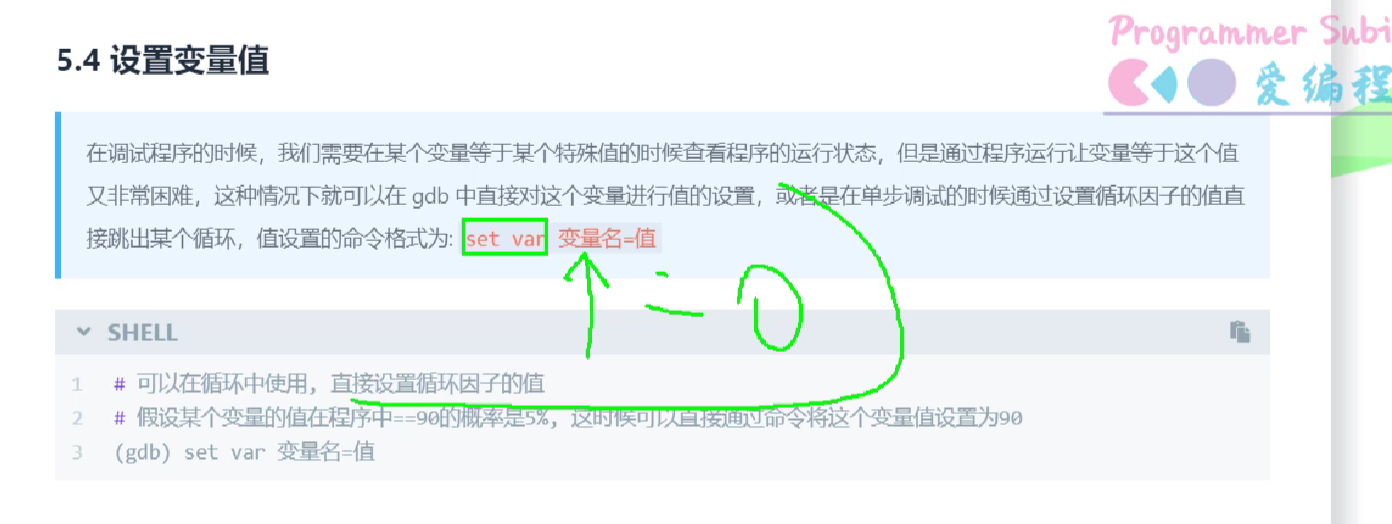


step (会进到函数题内部)

finish (如果想要跳出函数必须保证函数体内不能有有效断点，否则无法跳出)

next

utill （跳出循环体） 如果循环体内部有断点则不会跳出



set var 变量名 = 值