### Day04

1. 说明函数传参的方式和异同

按值传递

使用这种方式，调用函数本身不对实参进行操作，也就是说，即使形参的值在函数中发生了变化，实参的值也完全不会受到影响，仍为调用前的值。这相当于把实参赋值一份给形参，然后调用后内存回收。

地址传递

地址传递与按值传递的不同在于，它把实参的存储地址传送给对应的形参，从而使得形参指针和实参指针指向同一个地址。因此，被调用函数中对形参指针所指向的地址中内容的任何改变都会影响到实参。

2、写出下面所描述的各个函数的ANSI函数头。注意:只写出函数头即可，不需要实现。

a) donut()接受一个 int 类型的参数，然后输出若干个0，输出0的数目等于参数的值。

void donut(int a)

b)gear()接受两个int类型的参数并返回int类型的值。

int gear(int a, int b)

c)stuff\_it()的参数包括一个double类型的值以及一个double类型变量的地址，功能是把第一数值存放到指定的地址中。

void stuff\_it(double a, double\* p)

1. 编写一个函数，使其返回 3 个整型参数中的最大值。

#include<stdio.h>

int Max(int a, int b, int c)

{

return (a>b)?((a>c)?a:c):(b>c)?b:c;

}

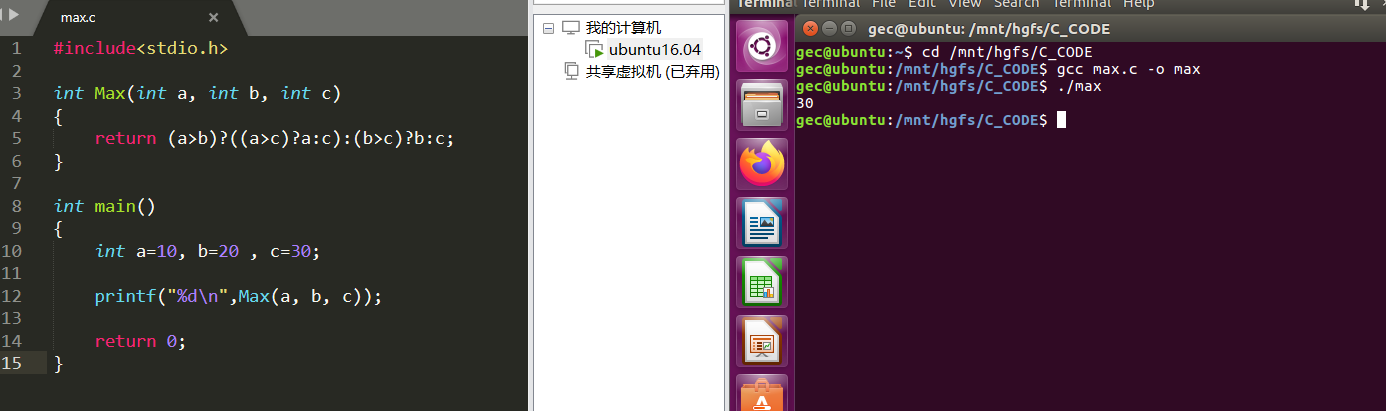
int main()

{

int a=10, b=20 , c=30;

printf("%d\n",Max(a, b, c));

return 0;



4、编写一个函数 my\_power，用循环的方法实现返回一个 float 类型数的某个整数次幂(保留 6 位小数)。例如： 调用 my\_power(3.14, -2)返回 0.101424

4、编写一个函数 my\_power，用循环的方法实现返回一个 float 类型数的某个整数次幂(保留 6 位小数)。例如： 调用 my\_power(3.14, -2)返回 0.101424

#include<stdio.h>

float my\_power(float num, int n)

{

int i=0,count=0;

float result = 1.0;

double value = 0;

if (n < 0) //判断正负次方

{

count = -1 \* n;

}

else

{

count = n;

}

for (i=1; i<=count; i++)

{

result \*= num;

}

if (n < 0) //负次方做出调整

{

value = 1.0 / result ;

}

return value;

}

int main()

{

printf("%.6lf\n",my\_power(3.14, -2));

return 0;

}



5、 编写一个程序，将两个字符串连接起来，不要用 strcat 或 strncat 函数。

#include <stdio.h>

#include <string.h>

char\* str\_connect(char\* a, char\* b, int len)

{

char\* p = a + len;

char\* q = b;

while (\*(b-1) == '\0')

{

\*p++ = \*q++;

}

//printf("%d\n",len);

return a;

}

int main()

{

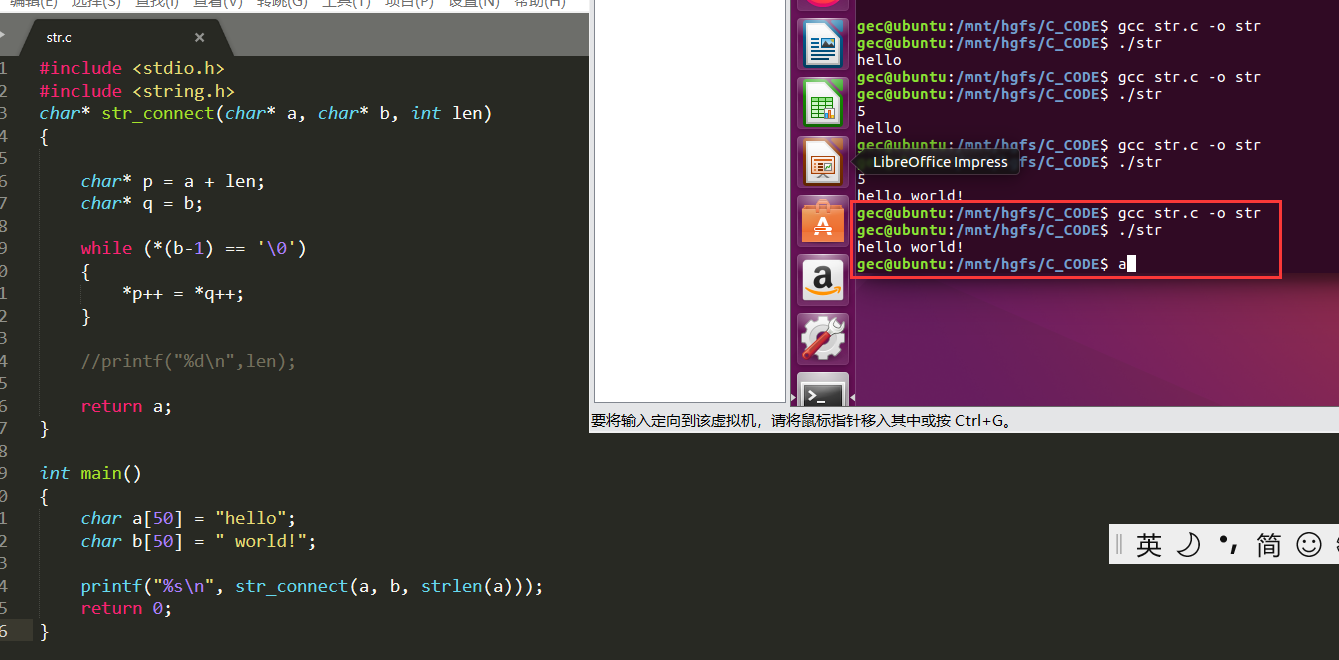
char a[50] = "hello";

char b[50] = " world!";

printf("%s\n", str\_connect(a, b, strlen(a)));

return 0;

}



6、 编写一个函数 Fibonacci()，要求程序输出第 n 项斐波那契数，n 由用户输入。 斐波那契数列：1, 1, 2, 3, 5, 8, 13, 21, … …

#include <stdio.h>

int fib(int num)

{

if (num == 1)

{

return 1;

}

else if (num == 0)

{

return 1;

}

else

{

return fib(num-1)+fib(num-2);

}

}

int main()

{

int n=0;

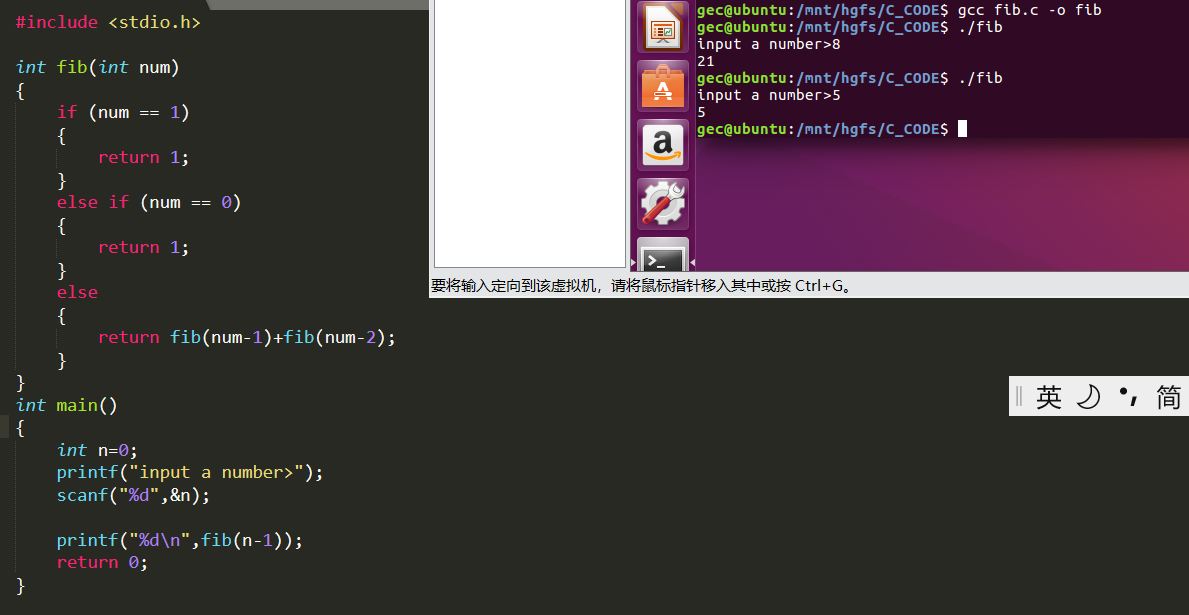
printf("input a number>");

scanf("%d",&n);

printf("%d\n",fib(n-1));

return 0;

}



7.编写一个程序，清除用户输入字符串中的空格符并将之输出。（例如用户输入”a b”，输出”ab”）

#include <stdio.h>

#include <string.h>

char\* Clear\_(char\* a, char\* b)

{

char\* p = a;

char\* q = b;

while (\*p!='\0')

{

while (\*p == ' ')

{

p++;

}

\*q++ = \*p++;

}

return b;

}

int main()

{

char a[50] = {0};

char b[50] = {0};

printf("input sentence>");

gets(a);

printf("%s\n", a);

printf("%s\n", Clear\_( a, b));

return 0;

}

