

```
main( int argc, char *argv[])
```

第一个参数是整形，第二个参数是指向字符的指针数组

特别的：

argv[0]为自身运行目录路径和程序名，

argv[1]指向第一个参数，

argv[2]指向第二个参数

首先arg是单词arguments（参数）的缩写

argc是argument counter（参数计数器）的缩写

argv是argument vector（参数矢量）的缩写

个人觉得第二个参数的命名很有魔性，因为矢量在物理学上指由大小和方向共同决定的量。

是一个任意起点到确定里程的终点之间连线，这与内存中只要给指针加一指针就向后跳一个指针所指向类型的长度，例如int*型的指针p，p+1,就向后跳了四个字节，char*型的指针p，p+1,就向后跳了一个字节

所以我们可以将指针理解成地址矢量，每种类型的指针都是对应类型的单位地址矢量，给Int*的指针加一就相当于给int*型的单位地址矢量乘2，于是int*的指针就往后走了四个字节，char*型的指针同理也如此。

main是整个程序的入口（就是程序的起点），main函数不能被其它函数调用，因此不可能在程序内部取得实际值。

main函数的参数值是从操作系统命令行上获得的。当我们要运行一个可执行文件时，在DOS提示符下键入文件名，再输入实际参数即可把这些实参传送到main的形参中去

具体操作如下：

首先在C++中完成编译之后，然后在dos命令窗口下切换到.exe执行文件所在的文件夹(dos下如何切换目录见DOS中切换盘符及目录)，再在该路径下输入带参数的命令：

XXX.exe a b c ...即可得到输出。

运行以下简单的代码

```
int main(int argc, char *argv[])
{
```

```

printf("argc = %d\n", argc);
for(int i=0; i<argc; ++i)
{
    printf("argc[%d] = %s\n", i, argv[i]);
}

return 0;
}

```

结果如下

参数个数默认为1

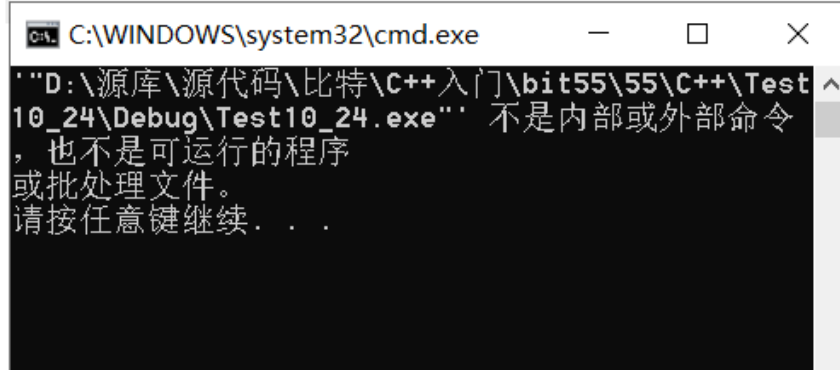
且为自身运行目录路径和程序名

```

int main(int argc, char *argv[])
{
    printf("argc = %d\n", argc);
    for(int i=0; i<argc; ++i)
    {
        printf("argc[%d] = %s\n", i, argv[i]);
    }

    return 0;
}

```



这中情况请参考：

https://blog.csdn.net/qq_43808700/article/details/102726840?utm_source=app