带参数的main函数

1.引入

可执行文件运行时，目前只能简单的运行，如果能加上参数，则使用中可以更灵活

例1：两数交换(常规方法)

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, t;
    cout << "请输入两个整数" << endl;
    cin >> a >> b;
    cout << "交换前：a=" << a << "  b=" << b << endl;
    t = a;
    a = b;
    b = t;
    cout << "交换后：a=" << a << "  b=" << b << endl;

    return 0;
}
```

带参数的main函数

2.方法

2.1.带参数的main函数的定义形式

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

两者均可

2.2.参数解释

argc：参数的个数，若不带参数，则为1(自身)

argv：参数的内容，用指针数组表示，每个元素是一个字符串(char *)，最后一个是 NULL

★ 参数名argc/argv可变，类型不能变（例如：int ac, char **av）

带参数的main函数

2.方法

2.3.使用

例2：两数交换(main函数带参数方法)

```
#include <iostream>
#include <cstdlib>          //atoi函数用到
using namespace std;
int main(int argc, char *argv[])
{
    int a, b, t;

    cout << "argc="   << argc    << endl;
    cout << "argv[0]="<< argv[0] << endl;
    a = atoi(argv[1]);     //atoi是将字符串转为整数的函数
    b = atoi(argv[2]);

    cout << "交换前:a=" << a << "  b=" << b << endl;
    t = a;
    a = b;
    b = t;
    cout << "交换后:a=" << a << "  b=" << b << endl;

    return 0;
}
```

假设编译后形成形成demo.exe

1、集成环境运行　　(出错,为什么？)

2、命令行运行

　　demo　　(出错,为什么？)

　　demo 10　　(出错,为什么？)

　　demo 10 15　　(正确)

　　demo 10 15 20　(正确)

带参数的main函数

2.方法

2.3.使用

例3：两数交换(main函数带参数方法 - 改进)

```
#include <iostream>
#include <cstdlib>          //atoi函数用到
using namespace std;
int main(int argc, char *argv[])
{   int a, b, t;
    if (argc<3) { /* 参数不足3个则出现提示 */
        cerr << "请带两个整数作为参数"<< endl;
        return -1;
        }
    for (t=0; t<argc; t++)    /* 打印所有的参数值 */
         cout << "argv[" << t << "]=" << argv[t] << endl;
    a = atoi(argv[1]);      //atoi是将字符串转为整数的函数
    b = atoi(argv[2]);
    cout << "交换前：a=" << a << "  b=" << b << endl;
    t = a;
    a = b;
    b = t;
    cout << "交换后：a=" << a << "  b=" << b << endl;
    return 0;
}
```

假设编译后形成形成demo.exe

1、集成环境运行

2、命令行运行

```
demo
demo 10
demo 10 15
demo 10 15 20
```

带参数的main函数

2.方法

2.4.综合应用

例4：作业相似度检查程序的参数设计

(1) 学生的匹配

要求能在两个特定的学生之间检查

某个特定学生和全体学生之间检查

全体学生之间相互检查

(2) 文件的匹配

要求既可以是单文件，也可以全部文件

(3) 相似度设置

要求值在60-100间浮动

(4) 输出方式

可选文件/屏幕

假设Linux下编译后形成形成 check，下列方式都正确

```
./check 1759999 1759998 7-b2.cpp 80 screen
./check 1759999 1759998 all      80 screen
./check 1759999 all     7-b2.cpp 75 result.txt
./check 1759999 all     all      85 screen
./check all     all     all      85 final.txt
```

★ 具体通过作业方式来理解实现过程

带参数的main函数

2.方法

2.5.参数个数不固定的带参main函数

例5：在Windows的命令行下输入 ping，可以看到ping 命令的很多选项，下列命令都是正确的

```
ping 10.60.102.252
ping -t 10.60.102.252
ping -n 10 10.60.102.252
ping -n 10 -l 50000 192.168.80.230
ping -t -l 50000 192.168.80.230
ping -l 50000 -t 192.168.80.230
```

★ 参数个数不固定，且部分参数要2个一组

★ 参数出现顺序任意

★ 具体通过作业方式来理解实现过程

带参数的main函数

2.方法

2.6.带参数的main函数的扩展形式(仅了解)

形式： int main(int argc, char **argv, char **env)

或： char *env[]

参数解释：
- argc：同前
- argv：同前
- env ：操作系统的环境变量，用指针数组来表示，每个元素是一个字符串(char *)，最后一个元素是NULL

使用：需要判断/取操作系统的某些设置时才用到

例6：取操作系统的环境变量

```
#include <iostream>
using namespace std;

int main(int argc, char **argv, char **env)
{
    int i;
    for (i=0; env[i]; i++)
        cout<< "env[" << i << "]=" << env[i] <<endl;

    return 0;
}
```