

在最终评测时，调用栈占用的空间大小不会有单独的限制，但在我们的工作环境中默认会有 **8 MB** 的限制。这可能会引起函数调用层数较多时，程序发生栈溢出崩溃。

我们可以使用一些方法修改调用栈的大小限制。例如，在终端中输入下列命令

```
ulimit -s 1048576
```

此命令的意义是，将调用栈的大小限制修改为 **1 GB**。

例如，在选手目录建立如下 **sample.cpp** 或 **sample.pas**

<i>sample.cpp</i>	<i>sample.pas</i>
<pre>void dfs(int a){     if(a == 0)         return;     int t = a;     dfs(a - 1); } int main(){     dfs(1000000);     return 0; }</pre>	<pre>procedure dfs(a: longint);     var t: longint;     begin         if a = 0 then             exit;         t := a;         dfs(a - 1);     end; begin     dfs(1000000); end.</pre>

将上述源代码编译为可执行文件 **sample** 后，可以在终端中运行如下命令运行该程序

```
./sample
```

如果在没有使用命令“`ulimit -s 1048576`”的情况下运行该程序，**sample** 会因为栈溢出而崩溃；如果使用了上述命令后运行该程序，该程序则不会崩溃。

特别地，当你打开多个终端时，它们并不会共享该命令，你需要分别对它们运行该命令。

请注意，调用栈占用的空间会计入总空间占用中，和程序其他部分占用的内存共同受到内存限制。