

Boas práticas de programação

Programação I
2020.2021

Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Reminder... Como programar?



Perceber o problema

Dados e Resultados

Pensar numa solução

Dividir o problema em problemas mais simples

Resolver os problemas mais simples

Resolver o problema mais complexo

Implementar a solução

Utilizar funções para estruturar a resolução dos sub-problemas

Testar a solução

Fazer vários testes

Escolher valores que induzam comportamentos diferentes do programa

Sumário

Características de um bom programa

Boas práticas

Características de um bom programa

Características de um bom programa

Perspectiva do utilizador

Satisfaz as especificações

Não contém erros e é robusto

Perspectiva do programador

Código inteligível

Fácil manutenção

Código

- Documentado
- Identado adequadamente
- Sem complexidade desnecessária
- Nomes explícitos
- ...

Exemplo 1

```
#define _ F-->00 || F-00--;
long F=00,00=00;
main(){F_00();printf("%1.3f\n", 4.*-F/00/00);}
F_00()
{
```

www.ioccc.org

Calcula o valor aproximado de pi!

Exemplo 2

```
#include <stdio.h>
main(t,_,a)
char *a;
{return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86, 0, a+1 )+a)):1,t<_?main(t+1, _, a ):3,main ( -94, -27+t, a
)&&t == 2 ?_<13 ?main ( 2, _+1, "%s %d %d\n" ):9:16:t<0?t<-72?main(_,
t,"@n'+,#'/*{ }w+/w#cdnr/+,{ }r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+\
,/+ #n+,/#;#q#n+,/+k#;*+,/'r : 'd*'3,{ }w+K w'K: '+'e#';dq#'l q#'+d'K#!/\
+k#;q#'r}eKK#}w'r}eKK{n l]' /#;#q#n') { }#}w') { } {n l]' /+ #n';d}rw' i;# ) {n\
l]! /n {n#'; r { #w'r nc {n l]' /# {l,+'K {rw' iK{;[{n l]' /w#q#\
n'wk nw' iwk{KK{n l]! /w{% 'l##w# ' i; : {n l]' /* {q# 'ld;r'} {n lwb!/*de}'c \
;;{n l' - { }rw]' /+,}## '*'#nc, ', #nw]' /+kd'+e}+;\
#'rdq#w! nr' / ' ) }+} {rl# ' {n' ' )# } '+'}## (!!/" )
:t<-50?_==*a ?putchar(a[31]):main(-65,_,a+1):main((*a == '/' )+t,_,a\
+1 ):0<t?main ( 2, 2 , "%s"): *a=='/' ||main(0,main(-61,*a, "!ek;dc \
i@bK'(q)-[w]*%n+r3#l,{ }:\nuwloca-0;m .vpbks,fxntdCeghiry"),a+1);}
```

www.iocccc.org

Um bom programa

Eficaz

Satisfaz as especificações do problema

Eficiente

Determina as soluções em tempo útil

Não contém erros e é robusto

Suporta bem as especificações incorretas por parte do utilizador

Legível e intelegível

Bem escrito, documentado e estruturado

Fácil manutenção

Fácil inserção de alterações, remoções e novas funcionalidades

Portável

Pode ser compilado noutras versões da linguagem e correr noutros sistemas operativos e ser facilmente convertido para outras linguagens

Boas práticas

Boas práticas

Não devemos aumentar, para além do necessário, o número de entidades requeridas para explicar qualquer coisa

William Occam (1284-1347)

A arte de programar é a arte de organizar a complexidade!

Edsger Dijkstra, “Programação Estruturada” (1972)

Outro exemplo

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{int pp, p = 0, t = 0;
float o, l, a = 0.0, aa, d = 1.0;
while (d > 1.0e-8){
pp = 1;
while(pp <= 5000){
o = (float) rand()/RAND_MAX;
l = (float) rand()/RAND_MAX;
if(o*o + l*l < 1) p++;
pp++;}
t = t + 5000;
aa = (float) p/t;
printf("Area para %d pontos: %f\n", t, aa);
d = fabs(aa - a);
a = aa;}
printf("Estimativa de pi: %6.3f\n", 4*a);
}
```

Boas práticas

1. Usar **indentação**: margens diferenciadas para as estruturas de controlo
2. Posicionar delimitadores das estruturas de forma a mostrar onde começam e acabam
3. Usar nomes sugestivos para as variáveis, reveladoras do seu papel.
4. Documentar o programa com comentários apropriados
5. Evitar a proliferação de variáveis
6. Evitar repetições de código – para isso existem **funções**!
7. Não misturar várias tarefas na mesma função
8. Se possível, evitar embutimentos de **if-else**
9. Instruções **if** precedendo ou dentro de ciclos e que testam a mesma condição, são, geralmente, desnecessárias
10. Remover todos os ciclos desnecessários
11. Eliminar todas as operações desnecessárias dentro de ciclos
12. Dentro de ciclos chamar apenas funções simples
13. Não usar truques
14. Não sacrificar inteligibilidade por eficiência