Programação Estruturada

Folha de exercícios 1

1999–2000

Definições “macro” e Pré-processador

Problema 1: Considere o programa seguinte:

```
#define X 100
#define DOBRO(A)  A+A
#define MAX(A,B) (A>=B ? A:B)

int main() {
  int a=10, b=20, c;
  a=b+X;
  c=DOBRO(a)*DOBRO(b);
  a=MAX(b,c);
  a=MAX(fact(b),fact(c));
}
```

1. Escreva como fica o programa após a expansão pelo pré-processador.
2. Utilizando o computador e, sempre que necessário, o manual do gcc, determine qual é de facto o programa após a expansão. Compare com as suas previsões (alínea anterior) e justifique.
3. Considere o mesmo programa com a linha inicial adicional “#include <stdio.h>. Determine o programa expandido. Onde se encontra (caminho completo) o ficheiro stdio.h?

Problema 2: Defina “macros” para realizar os objectivos indicados. Para cada caso teste, com um pequeno programa, a “macro” que construiu.

1. PAR(x): Será que que inteiro x é par?
2. DUPLICA(x): Dobro de x (utilize o operador <<).
3. LETRA(x): Será x é uma letra (maiúscula ou minúscula)?
4. DIGITO(x): Será x é um dígito?
5. HEXA(x): Será x é um algarismo hexadecimal (0 . . . 9, 'a'. . . 'h', 'A'. . . 'H')?
6. MAIOR(a,b): o maior entre dois valores a e b.

Problema 3: Escreva um programa que leia caracteres da entrada-padrão e filtre todos os que são letras (maiúsculas ou minúsculas) ou dígitos e os escreva da saída-padrão.

Ou seja: todos os caracteres lidos que estejam nessas condições (e só esses) devem ser impressos. Teste o seu programa com: (i) o próprio texto C do seu programa, (ii) o executável do seu programa.

Problema 4: Escreva um programa que leia caracteres da entrada-padrão e converta para minúsculas todas as letras maiúsculas. Todos os outros caracteres devem ficar inalterados.

Problema 5: Sendo `filtra` e `minusc` os executáveis correspondentes respectivamente aos dois problemas anteriores, experimente e explique o efeito da linha de comandos UNIX:

```
filtra < fich | minusc
```

onde `fich` é um ficheiro qualquer.

Problema 6: Como é que o pré-processador expande a seguinte definição "macro"? Experimente e tire uma conclusão...

```
#define factorial(n)  (n*factorial(n-1))
```

Problema 7: Crie dois ficheiros de texto `yin.h` e `yang.h` com o seguinte conteúdos:

yin.h: `#include "yang.h"`

yang.h: `#include "yin.h"`

Crie um ficheiro `zen.c` com a linha `#include "yin.h"` (ou `yang.h`, tanto faz); como é que o pré-processador expande esse ficheiro? Experimente e tire uma conclusão...