

Ficha 1

Programação Imperativa

1 Estado e atribuições

Diga, justificando, qual o output de cada um dos seguintes excertos de código C.

- ```
int x, y;
x = 3; y = x+1;
x = x*y; y = x + y;
printf("%d_%d\n", x, y);
```
- Assuma que os códigos ASCII dos caracteres 'A', '0', ' ' e 'a' são respectivamente 65, 48, 32 e 97)  

```
char a, b, c;
a = 'A'; b = '_'; c = '0'
 ,;
printf ("%c_%d\n", a, a)
 ;
a = a+1; c = c+2;
printf ("%c_%d_%c_%d\n",
 a, a, c, c);
c = a + b;
printf ("%c_%d\n", c, c)
 ;
```

- ```
int x, y;  
x = 0;  
printf ("%d_%d\n", x, y)  
    ;
```
- ```
int x, y;
x = 200; y = 100;
x = x+y; y = x-y; x = x-
 y;
printf ("%d_%d\n", x, y)
 ;
```

### 2 Estruturas de controlo

1. Diga, justificando, qual o output de cada um dos seguintes excertos de código C.

- ```
int x, y;  
x = y = 0;  
while (x != 11) {  
    x = x+1; y += x;  
}  
printf ("%d_%d\n", x, y)  
    );
```

- ```
int x, y;
x = y = 0;
while (x != 11) {
 x = x+2; y += x;
}
printf ("%d_%d\n", x, y)
);
```

```

• int i;
 for (i=0; (i<20) ; i++)
 if (i%2 == 0) putchar
 ('_');
 else putchar ('#');

• char i, j;
 for (i='a'; (i != 'h');
 i++) {
 for (j=i; (j != 'h');
 j++)
 putchar (j);
 putchar ('\n');
 }

•
void f (int n) {
 while (n>0) {
 if (n%2 == 0) putchar
 ('0');
 else putchar ('1');
 n = n/2;
 }
 putchar ('\n');
}

int main () {
 int i;
 for (i=0; (i<16); i++)
 f (i);
 return 0;
}

```

2. Escreva um programa que desenhe no ecran (usando o caracter #) um quadrado de dimensão 5. Defina para isso uma função que desenha um quadrado de dimensão **n**. Use a função **putchar**. O resultado da invocação dessa função com um argumento 5 deverá ser o que se apresenta à direita.

```

#####
#####
#####
#####
#####

```

3. Escreva um programa que desenhe no ecran (usando os caracteres # e \_) um tabuleiro de xadrez. Defina para isso uma função que desenha um tabuleiro de xadrez de dimensão **n**. Use a função **putchar**. O resultado da invocação dessa função com um argumento 5 deverá ser o que se apresenta à direita.

```

#_#_#
##_
#_#_#
##_
#_#_#

```

4. Escreva duas funções que desenharam triângulos (usando o caracter #). O resultado da invocação dessas funções com um argumento 5 deverá ser o que se apresenta à direita.

```

#
###
#####
#####
#####
#####
####
###
##
#

```

Defina cada uma dessas funções (com o nome **triangulo**), num ficheiro separado (**vertical.c** e **horizontal.c**). Compile esses dois ficheiros (usando o comando **gcc -c**) separadamente.

Considere agora o programa `triangulo.c` ao lado.

Compile este programa (com o comando `gcc -c triangulo.c`). Construa (e use) agora dois executáveis, usando os comandos

- `gcc -o t1 triangulo.o vertical.o`
- `gcc -o t2 triangulo.o horizontal.o`

```
#include<stdio.h>

void triangulo (int n)
 ;

main () {
 triangulo (5);
 return 0;
}
```