

Algoritmos e Estruturas de Dados

Ficha 3

Ano Lectivo de 2022/23

1 Estado e atribuições

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

(a)

```
int x,y;
x = 3; y = x +1;
x = x*y; y = x+y;
printf("%d %d\n", x, y);
```

(b)

```
int x, y;
x = 0;
printf("%d %d\n", x, y);
```

- (c) (assuma que os códigos ASCII dos caracteres 'A', '0', ' ' e 'a' são respetivamente 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);
```

(d)

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

(e)

```
int x, y;
x = 100; y = 28;
x +=y; y -=x;
printf("%d %d\n", 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 em C.

(a)

```
int x, y;
x = 3; y = 5;
```

```

    if (x > y)
        y = 6;
    printf("%d %d\n", x, y);

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

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

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

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

(f) 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 ecrã (usando o caracter #) um quadrado de dimensão 5. Defina para isso uma função que desennha 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 abaixo.

```
#####
#####
#####
#####
#####
```

3. Escreva um programa que desenhe no ecrã (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 argumento 5 deverá ser o que se apresenta abaixo.

```
#_#_#
_#_#_
#_#_#
_#_#_
#_#_#
```

4. Escreva duas funções que desenham triângulos (usando o carácter #). O resultado da invocação dessas funções com um argumento 5 deverá ser o seguinte,

```
#                #
##              ###
###            #####
####          #####
#####        #####
#####
####
###
##
#
```

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 problema *triangulo.c* abaixo.

```
#include<stdio.h>

void triangulo (int n);

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

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*