



Escola de Engenharia da Universidade do Minho
Mestrado Integrado em Eng. Electrónica Industrial e Computadores
Programação de Computadores

2012/2013
MIEEIC
(1º Ano)
1º Sem

Exame 2 - Data 25/01/2013, Duração 1h45m

Nome: _____ N° Aluno _____

Responda às seguintes questões, preenchendo a tabela com a **opção correcta (em maiúsculas)** (Correcto: x Val / Errado: -x/3 Val).
Suponha que foram realizados as inclusões das bibliotecas necessárias (exemplo: #include <stdio.h> e #include <stdlib.h>).

1	2	3	4	5	6	7	8	9	10	11	12

Suponha a função em linguagem C:

```
void f1(int a, int b)
{
    if(b>5 && a>b || !(a>5)) printf("4");
    else if(b>6 || a<3) printf("3");
    else printf("2");
    printf("1");
}
```

1) Se executar a chamada f1(7,6) o que é escrito no ecrã?

- a) 1 b) 41 c) 31 d) 21
e) Nenhuma das Anteriores

2) Se executar a chamada f1(1,8) o que é escrito no ecrã?

- a) 1 b) 41 c) 31 d) 21
e) Nenhuma das Anteriores

Suponha a seguinte função em linguagem C/C++:

```
void ciclos(int a, int b)
{
    for(int i=1; i<a-1; i++) {
        for(int j=2; j<b+1; j++)
            if(i==j) printf("#");
            else printf("0");
        printf("\n");
    }
}
```

3) Executando ciclos(6,6) o que é escrito no ecrã?

- a) b) c) d)
#000 00000 #0000 #####
0#00 #0000 0#000 00000
00#0 0#000 00#00 00000
000# 00#00 000#0 00000
0000

e) Nenhuma das Anteriores

4) Executando ciclos(3,5) o que é escrito no ecrã?

- a) b) c) d)
#00 000 #000 0000
0#0

e) Nenhuma das Anteriores

Suponha a função xpto:

```
void xpto(int a) {
    for(int i=0; i<10; i++)
        printf("%c ", i+a);
}
```

5) O que escreve no ecrã xpto(48) ?

- a) Escreve as letras maiúsculas A até J
b) Escreve as letras minúsculas a até j
c) Escreve os números de 48 até 57
d) Escreve os números de 0 até 9
e) Nenhuma das Anteriores

6) Indique o que escreve no écran xpto(48) substituindo
printf("%c ", i+a); por printf("%d ", i+a);?

- a) Escreve as letras maiúsculas A até J
b) Escreve as letras minúsculas a até j
c) Escreve os números de 48 até 57
d) Escreve os números de 0 até 9
e) Nenhuma das Anteriores

7) Supondo a seguinte função e programa em linguagem C
indique o que é escrito no ecrã.

```
#define DI 7
void filtra(double v[], double f[]) {
    for(int i=1; i<DI-1; i++)
        f[i]=(v[i-1]+v[i]+v[i+1])/3.0;
    f[0]=(v[0]+v[1])/2.0;
    f[DI-1]=(v[DI-2]+v[DI-1])/2.0;
}
int main(void)
{
    double fil[DI];
    double vec[DI]={1,2,3,4,2,9,1};
    filtra(vec, fil);
    for(int i=0; i<DI; i++)
        printf("%1.1f ", fil[i]);
}
```

- a) 1.5 2.0 3.0 3.0 4.0 3.0 3.5
b) 1.5 2.0 3.0 3.0 5.0 4.0 3.5
c) 1.0 2.0 3.0 4.0 2.0 6.0 1.0
d) 1.5 2 3 3 4 3 3.5
e) Nenhuma das Anteriores

8) Indique o que é escrito no ecrã substituindo double
vec[DI] = {1,2,3,4,2,9,1}; por double
vec[DI] = {1,4,1,4,1,4,1};

- a) 2.0 3.0 2.0 3.0 2.0 3.0 2.0
b) 3.0 3.0 3.0 3.0 3.0 3.0 3.0
c) 3.0 2.0 3.0 2.0 3.0 2.0 3.0
d) 1.0 4.0 1.0 4.0 1.0 4.0 1.0
e) Nenhuma das Anteriores

9) Indique o que faz o seguinte programa:

```
int main(void)
{
    int n;
    FILE *fich = fopen("t.txt", "w");
    if (fich!=NULL) {
        scanf("%d", &n);
        for(int i=1; i<=n; i++) {
            for(int j=1; j<=10; j++)
                fprintf(fich, "%3dx%3d =%3d\n",
                    i, j, i*j);
            fprintf(fich, "\n");
        }
        fclose(fich);
    }
}
```

- a) Lê um ficheiro com a tabuada dos n (dado pelo utilizador)
- b) Imprime no ecrã a tabuada dos n (dado pelo utilizador)
- c) Cria um ficheiro com a tabuada dos n (dado pelo utilizador)
- d) Imprime no ecrã as tabuadas entre 1 e n (n dado pelo utilizador)
- e) Nenhuma das Anteriores

10) Indique o que escreve no ecrã a seguinte função:

```
int compara(void)
{
    char s1[]="bXXX"; char s2[]="bXXX";
    char s3[]="XBBB"; char s4[]="0000";
    printf("%d %d %d \n", strcmp(s1,s2),
        strcmp(s1,s3), strcmp(s1,s4));
}
```

- a) 0 -1 1
- b) 0 -1 -1
- c) 0 1 1
- d) -1 -1 -1
- e) Nenhuma das Anteriores

11) Indique o que escreve no ecrã a seguinte função:

```
void fstring(void) {
    char st[] = "MundialFutebol";
    printf("%c-%d\n",
        st[strlen(st)-1], strlen(st));
}
```

- a) i-7 b) 14-1 c) 1-14 d) o-13
- e) Nenhuma das Anteriores

12) Indique o que escreve no ecrã o seguinte programa:

```
struct pos {
    int x; int y;
};

void func(pos &a) {
    a.x *=2; a.y *= 2;
}

int main(void) {
    pos p1; p1.x=2; p1.y=3;
    func(p1);
    printf("%d %d\n", p1.x, p1.y);
}
```

- a) 2 3
- b) 2 2
- c) 4 6
- d) Não escreve nada pois dá erro de compilação
- e) Nenhuma das Anteriores

13) Escreva uma função void verifica_num(int num, int x); que verifique (e imprima no écran) se um dado número inteiro (num) é par e ao mesmo tempo se é múltiplo de x, se é apenas par, se é apenas múltiplo de x ou nenhum dos dois. Exemplos:

```
verifica_num(20, 5)
O número 20 é par e é múltiplo de 5.
verifica_num (25, 5)
O número 25 não é par e é múltiplo de 5.
verifica_num (8, 5)
O número 8 é par e não é múltiplo de 5.
verifica_num (13, 5)
O número 13 não é par e não é múltiplo de 5.
```

14) Construa a função void letZ(int dim, char ch) que dada a dimensão dim, desenhe letras "Z" compostas de símbolos ch com dimensão dim, de acordo com os exemplos apresentados. Na resolução do exercício deve usar a seguinte função que já se encontra implementada:

```
void nchar(int n, char c) {
    for(int i=0; i<n; i++) printf("%c", c);
}

letZ(1, '*') letZ(3, '*') letZ(6, 'O') letZ(7, 'Z')
*          ***          OOOOOO      ZZZZZZZ
          *              O              Z
          ***            O              Z
                  O          Z
                  O          Z
                  OOOOOO      Z
                          ZZZZZZZ
```

15) Escreva uma função `int vec_ordenado(int vec1[], int Dim, int n1, int n2);` que verifica se os elementos entre o elemento `n1` e o elemento `n2` (inclusive) do vector `vec1` estão ordenados por ordem crescente. Suponha que o primeiro elemento do vetor é o elemento 0 e último é o elemento `Dim-1`. A função deve retornar 1 em caso afirmativo e 0 caso contrário.

16) Suponha um ficheiro contendo nomes próprios, um por cada linha. Implemente uma função `int nomes(char nom1[], char nom2[]);` que dado o ficheiro "nomes.txt" e dados dois nomes próprios `nom1` e `nom2` retorne qual dos nomes (1 ou 2) tem maior número de ocorrências no ficheiro. Em caso de igualdade a função deve retornar o número do nome (1 ou 2) que alfabeticamente aparece primeiro.

17) Suponha uma matriz representando o tabuleiro de um jogo contendo números inteiros representando peças de dois jogadores (1 e 2), espaços vazios (0), num tabuleiro com dimensão `NxN`. Suponha que uma jogada consiste em expandir uma peça na diagonal de `Dim` casas (ocupando todas as casas até ao limite de `Dim` casas em todas as diagonais). A ocupação de casa verifica-se até encontrar uma peça adversária que será capturada (não continuando mais nessa direção), uma peça do próprio jogador, ou o limite do tabuleiro.

Implemente a função `bool jogada(int jog, int lin, int col, int dim, int tab[N][N])` que efetua uma jogada, do jogador `jog` na posição `col, lin` com a dimensão `Dim`. Caso a jogada não seja possível dado que não existe uma peça do jogador na posição respetiva, a função deve retornar falso. Caso seja possível (existe uma peça do jogador `jog` em `lin, col`) deve retornar verdadeiro.

Exemplo, supondo o seguinte tabuleiro (`tab`) inicial com `N=6` (à esquerda), aplicando `jogada(1, 2, 4, 2, tab)`, `tab` fica com a configuração apresentada ao centro. Aplicando `jogada(2, 6, 3, 3, tab)`, resulta o tabuleiro representando à direita. Ambas as chamadas retornam verdadeiro dada a jogada ser possível.

1 0 0 0 0 0	1 0 1 0 1 0	1 0 0 0 0 0
0 0 0 1 1 0	0 0 0 1 1 0	0 0 0 1 1 0
0 0 0 0 2 0	0 0 1 0 1 0	0 0 0 0 2 2
0 0 0 1 0 0	0 1 0 1 0 0	0 0 0 1 2 0
0 1 0 0 0 0	0 1 0 0 0 0	0 2 0 2 0 0
0 0 2 0 0 0	0 0 2 0 0 0	0 0 2 0 0 0

18) Escreva uma função que simule o lançamento de dois dados de seis faces e retorne os respectivos valores `void lancamento_dados(int &v1, int &v2)`. Baseado nesta função, escreva uma função `void jogadasN(int nj, int jog, int tab[N][N])` que efetua `nj` jogadas válidas do jogador `jog`, no tabuleiro da alínea anterior.

19) Construa um programa principal que lhe permita testar, com exemplos relevantes, os programas construídos nas alíneas 13, 14, 15, 16, 17 e 18.