



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

2013/2014
MIEEIC
(1º Ano)
1º Sem

Exame 1, Duração 1h45m - Com Consulta

Nome: _____ Nº Alun.: _____

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	13	14	15

Suponha a seguinte função em linguagem C:

```
void f1(int a, int b)
{
    if(a>8 || a<b && a<6) printf("1");
    if(b>6||a<3) printf("2");
    else if (!(a>b))printf("3");
}
```

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

- a) 1 b) 2 c) 3 d) 13
e) Nenhuma das Anteriores

2) Se executar a chamada f1(5, 6) o que é escrito no ecrã?

- a) 1 b) 2 c) 3 d) 13
e) Nenhuma das Anteriores

Suponha a seguinte função em linguagem C:

```
void f2(int op)
{
    switch(op) {
        case 1: printf("1");
        case 2: case 3: printf("3");
        case 4: printf("4"); break;
        default: printf("5");
    }
}
```

3) Se executar a chamada f2(1) o que é escrito no ecrã?

- a) 1 b) 123 c) 1234 d) 5
e) Nenhuma das Anteriores

4) Se executar a chamada f2(6) o que é escrito no ecrã?

- a) 45 b) 4 c) 5 d) 6
e) Nenhuma das Anteriores

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

```
void imp(int x, int tip)
{
    do {
        printf("%d", x);
        if (tip==0) x++; else x*=2;
    } while(x<10);
}
```

5) Executando imp(5, 0) o que é escrito no ecrã?

- a) 12345
b) 12345678910
c) 5678910
d) 56789
e) Nenhuma das Anteriores
6) Executando imp(1, 1) o que é escrito no ecrã?

a) 12345678910

b) 123456789

c) 1248

d) 2468

e) Nenhuma das Anteriores

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

```
void ciclos(int n1, int n2)
{
    for(int i=1; i<n1; i++)
        for(int j=i+5; j>n2; j--)
            printf("%d-%d ", i, j);
}
```

7) Executando ciclos(2, 5) o que é escrito no ecrã?

- a) 1-6
b) 1-6 1-5 2-7 2-6 2-5
c) 1-6 1-5 1-4 2-7 2-6 2-5 2-4
d) 2-5 2-4
e) Nenhuma das Anteriores

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

- a) 3-3 4-4
b) 1-6 1-5 2-7 2-6 2-5
c) 1-6 1-5 1-4 2-7 2-6 2-5 2-4
d) 3-3 3-2 3-1 2-3 2-2 2-1
e) Nenhuma das Anteriores

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

```
void ciclos2(int N, int N2)
{
    for(int i=1; i<=N; i++) {
        for(int j=1; j<N2; j++)
            if(i<j) printf("#");
            else printf("O");
        printf("\n");
    }
}
```

9) Executando ciclos2(4, 3) o que é escrito no ecrã?

- | | | | |
|----|----|-----|------|
| a) | b) | c) | d) |
| O# | OO | O## | O### |
| OO | OO | OO# | OO## |
| OO | OO | OOO | OOO# |
| | OO | OOO | OOOO |

e) Nenhuma das Anteriores

10) Executando ciclos2(2, 2) o que é escrito no ecrã?

- | | | | |
|----|----|----|----|
| a) | b) | c) | d) |
| O | # | ## | O |
| | | | O |

e) Nenhuma das Anteriores

11) Supondo a seguinte função `xpto1` indique qual das seguintes afirmações está correta.

```
int xpto1()
{
    int v[10];
    for(int i=0; i<=9; i++)
        v[i]= i*2;
    for(int i=0; i<=9; i++)
        printf("%d ",v[9-i]);
    return v[5];
}
```

- a) Antes de executar a última instrução da função, o conteúdo do vetor `v` é {20, 18, 16, 14, 12, 10, 8, 6, 4, 2}
- b) Antes de executar a última instrução da função, o conteúdo do vetor `v` é {18, 16, 14, 12, 10, 8, 6, 4, 2, 0}
- c) A função retorna o valor 5
- d) A função retorna o valor 8
- e) Nenhuma das Anteriores

12) Indique o que é escrito no ecrã pela função anterior.

- a) 20 18 16 14 12 10 8 6 4 2
- b) 18 16 14 12 10 8 6 4 2 0
- c) 2 4 6 8 10 12 14 16 18 20
- d) 0 1 2 3 4 5 6 7 8 9
- e) Nenhuma das Anteriores

13) Supondo a seguinte função em linguagem C/C++ indique o que é escrito no ecrã.

```
void ap1(void){
    int *ap, x, y, z;
    x = 10; y=10; ap = &y;
    y = 2*x; z = 2*(*ap); *ap = 20;
    printf("x=%d y=%d z=%d \n",
        x,y,z);
}
```

- a) x=10 y=20 z=20
- b) x=10 y=20 z=40
- c) x=10 y=10 z=20
- d) x=20 y=20 z=40
- e) Nenhuma das Anteriores

14) Supondo o seguinte programa em linguagem C/C++ indique o que é escrito no ecrã.

```
void xpto(int *a, int *b, int *c){
    a = 1; b = 2; c = 3;
}
int main(void) {
    int x = 5, y = 10, z = 20;
    xpto(&x, &y, &z);
    printf("%d %d %d", x, y, z);
}
```

- a) 1 10 20
- b) 5 2 3
- c) 1 2 3
- d) Nada pois o programa dá erro na compilação
- e) Nenhuma das Anteriores

15) Diga o valor das variáveis `x`, `y`, `z` e `w` depois da seguinte sequência de operações:

```
int x=10,y,z,w;
y=x++; z=++x; x =++z; w=x++;
```

- a) x=14 y=11 z=12 w=12
- b) x=14 y=10 z=13 w=13
- c) x=12 y=11 z=13 w=13
- d) x=13 y=10 z=12 w=12
- e) Nenhuma das Anteriores

16) Construa uma função `void imprime_let(void)` que imprima no ecrã os códigos e símbolos de todas as letras minúsculas e maiúsculas (uma por linha e formatadas como se segue). Nota: no exemplo os ‘.’ representam espaços. Exemplo: `imprime_let()`

```
.97.->.a.|.65.->.A
.98.->.b.|.66.->.B
.99.->.c.|.67.->.C
100.->.d.|.68.->.D
...
121.->.y.|.89.->.Y
122.->.z.|.90.->.Z
```

17) Construa a função `void let_c(int alt, int larg)` que dada a altura `alt` e a largura `larg` desenhe letras C compostas por símbolos ‘X’ e ‘.’ com aspeto semelhante ao apresentado abaixo, de acordo com os exemplos apresentados:

<code>letc(1,1)</code>	<code>letc(2,2)</code>	<code>letc(3,1)</code>	<code>letc(1,3)</code>	<code>letc(3,2)</code>
X	XX	X	XXX	XX
	XX	X		X.
		X		XX
<code>letc(3,3)</code>		<code>letc(3,5)</code>		<code>letc(4,4)</code>
XXX		XXXXX		XXXX
X..		X....		X...
XXX		XXXXX		X...
				XXXX

Suponha que dispõe da seguinte função já implementada (e que deve utilizar sempre que for útil):

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

18) Construa uma função `double mensalidade(int tipo, int idade)` que dado o tipo de cliente e a sua idade retorne e imprima no ecrã a mensalidade que este deve pagar. Para clientes com menos de 18 anos de idade, ou séniores (idade igual ou superior a 65 anos), a mensalidade é metade da mensalidade dos adultos. A tabela de mensalidades para adultos é a seguinte: tipo 1: 10.50 Eur, tipo 2: 15.00 Eur, tipo 3: 18.30 Eur, tipo 4: 20.00 Eur, tipo 5: 23.20 Eur, Outro Tipo: 30.00 Eur. No entanto, devemos tomar em consideração que a bonificação de pagamento de metade da cota só se aplica a clientes dos tipos 1, 2, 3 e 4. Nota: Utilize a instrução `switch case`. Exemplos:

<code>mensalidade(2, 16)</code>	<code>mensalidade(8, 68)</code>
Mensalidade = 7.50 Eur	Mensalidade = 30.00 Eur

19) Construa uma função `int par_ou_impar(int vpar[], int *np, int vimp[], int *ni)` que permita ao utilizador introduzir elementos inteiros positivos e coloque todos os numeros impares introduzidos no vetor `vimp[]` e todos os elementos pares no vetor `vpar[]`. A entrada de valores termina quando o utilizador introduzir o valor -1. Deve ainda calcular o total de numeros pares (`*np`) e impares (`*ni`) introduzidos. Por fim, a função deve retornar o número total valores inválidos itroduzidos. Por exemplo, supondo que os seguintes números são introduzidos pelo utilizador:

```
3 -5 0 10 6 3 4 5 4 2 8 -5 1 -1
```

`vpar[]={10, 6, 4, 4, 2, 8}`, `vimp[]={3, 3, 5, 1}`, `*np = 6`, `*ni = 4`. A função, no final retorna 3 (dado que foram introduzidos três valores inválidos -5, 0 e -5).

20) Suponha uma matriz, já inicializada `tab[N][N]` contendo números inteiros representando peças e damas de dois jogadores (1 - peça jogador 1, 2 - peça jogador 2, 3 - dama do jogador 1, 4 - dama do jogador 2) e espaços vazios (0) num tabuleiro com dimensão `NxN`. Construa a função `int move_pec_Jog(int tab[N][N], int jog, int dir)` que dada a matriz com o estado do jogo movimente todas as peças e damas isoladas (i.e. sem nenhuma peça/dama de nenhum jogador na vizinhança), do jogador `jog` na direção `dir` (em que `dir = 1`: direita; `dir =2`: esquerda; `dir = 3`: cima e `dir=4`: baixo). As peças só se movimentam caso estejam inicialmente isoladas, a casa de destino esteja dentro do tabuleiro e livre. A função deve retornar o número de movimentos realizados.

Exemplo: Aplicando a função `move_pec_Jog(tab, 1, 2)`, à matriz inicial em baixo, é retornado o valor 2 (2 peças movimentadas para a esquerda) e muda a matriz para a matriz representada como resultado final:

Matriz Inicial:	Resultado Final:
1 1 0 0 1	1 1 0 1 0
1 4 0 1 0	1 4 0 1 0
0 0 0 2 0	0 0 0 2 0
1 0 2 0 3	1 0 2 3 0
0 0 3 0 0	0 0 3 0 0

21) Construa um programa principal que lhe permita testar, com exemplos relevantes, os programas construídos nas alíneas 16, 17, 18, 19 e 20.