

### Ficha de Avaliação Sumativa

EFA Tecnológico - Técnico de Informática e Sistemas

UFCD:0784 – Programação em C – C++ - Funções e Estruturas

Duração: 120 minutos

Nome Formando: Trigo Prats

Data: 23/01/12

Formador: João Leitão

Classificação: BOM 16,3 VALORES

### Notas Gerais:

Esta prova de natureza teórico-prática divide-se em três grupos e tem a duração de 120 minutos sem tolerância.

Utilize apenas caneta ou esferográfica de tinta indelével, azul ou preta.

Não é permitido o uso de corrector. Em caso de engano, deve riscar, de forma inequívoca, aquilo que pretende que não seja classificado.

Não é permitido o uso de auxiliares.

As respostas ilegíveis ou que não possam ser identificadas são classificadas com zero pontos.

Rubrique no canto superior direito todas as folhas da prova.

Qualquer tentativa de cópia será punida com a anulação da prova.

As cotações dos itens encontram-se descritas no quadro seguinte:

Grupo	Questão	Alínea	Cotação (Pontos)	TOTAL
Grupo I	1.		0,5	2
	2.		0,5	
	3.		0,5	
	4.		0,5	
Grupo II	1.		X 3	6
	2.		X 3	
Grupo III	1.		1,5	12
	2.		X 2,5	
	3.		X 2,5	
	4.		1,5	
	5.		X 4	
				20

**Atenção!** Se apresentar mais de uma resposta, ou resposta ambígua, a questão será anulada.

1. Indique, para cada afirmação, se é verdadeira ou falsa, justificando no caso de ser falsa:

a) Na declaração de uma função é obrigatório colocar o tipo de dados dos argumentos dessa função.

*sempre*  
~~É obrigatório colocar os tipos de dados~~  
~~ao declarar uma função.~~ *Se for Inteiro*  
*e inteiro, se for char e char um Tem*  
*de estar*

b) É possível no momento da declaração de uma função, não atribuir o nome da função.

~~É possível podem ficar somente~~  
~~as chavetas~~ *→ Argumentos*

c) Só é possível definir no máximo dois argumentos na declaração de uma função.

~~Não, ate podem ser mais~~  
~~Tem e de ser sempre o mesmo~~  
~~numero 3 para 3, 2 para 2~~  
~~etc~~

d) → Na chamada de uma função é obrigatório colocar o tipo de dados dos argumentos dessa função.

~~Não, na chamada de uma função~~  
~~não é obrigatório por o tipo de pode-se~~  
~~meter somente a quantidade~~  
~~de variáveis, obrigatório na declaração~~

## Grupo II

1. Indique qual o resultado final das variáveis **x**, **y**, **z**, **a** e **b** após a execução do seguinte fragmento de código:

```
int main(){
    int x;
    int y;
    int z;
    x=0;
    y=1;
    z=0;
    int a;
    int b;
    int c;

    if(x==0){
        a=2;
        b=x+y;
    }
    else if(x==0 || y>2){
        a=0;
        b=-1;
    }
    else{
        a=1;
        b=10;
    }

    c=funcao1(a,b);
    c=c+(a-b);
    x=x+1;
    y=y-1;
    z=z+1;
    printf("%d\n",x);
    printf("%d\n",y);
    printf("%d\n",z);
    printf("%d\n",a);
    printf("%d\n",b);
    printf("%d\n",c);

    return(0);
}

int funcao1(int b, int a){
    int res;
    res=b-a;
    res=res+(2*b);
    return(res);
}
```

x	y	z	a	b	c
0	1	0	2	1	3
1	0	1	1	2	

if(x==0)

a = 2

c = 3

x = 0

y = 1

z = 0

b = x + y = 0 + 1 = 1 ✓

c =

b	A	res
1	0	1
		3

C = ~~3~~   
 PASSA PARA b 4

~~res~~ = b - a = 1 - 0 = 1 ✓

res = 1 + (2 \* 1) = 3 ✓

~~res~~ = 1 + 2 = 3  
C = 3 + (2 - 1) = 3 - 1 = 2

P - x = 0 + 1 = 1

P - y = 1 - 1 = 0

P - z = 0 + 1 = 1

P - a = 2

P - b = 1

P - C = 26

Resolução pergunta 1 Grupo II

2. Indique qual o resultado final das variáveis: q, w, e, r, t, y e i após a execução do seguinte fragmento de código:

```
int main(){
```

```
int q;
```

```
int w;
```

```
int e;
```

```
int r;
```

```
int t;
```

```
int y;
```

```
int i;
```

```
q=5;
```

```
w=3;
```

```
e=4;
```

```
r=q+w;
```

```
t=(r+e)-q;
```

```
y=t-r;
```

```
if(q==0 && y<1){
```

```
    r=6;
```

```
    t=1;
```

```
    y=2;
```

```
else if(q==0 || t>=1){
```

```
    r=3;
```

```
    t=1;
```

```
    y=2;
```

```
else{
```

```
    r=1;
```

```
    t=5;
```

```
    y=1;
```

```
for(i=11;i>=3;i=i-2){
```

```
}
```

```
q=funcao1(i,r,t);
```

```
y=q*2;
```

```
t=funcao2(q);
```

```
printf("%d",q);
```

```
printf("%d",w);
```

```
printf("%d",e);
```

```
printf("%d",r);
```

```
printf("%d",t);
```

```
printf("%d",y);
```

```
printf("%d",i);
```

```
return(0);
```

```
}
```

```
int funcao1(int x, int y, int z){
```

```
int res;
```

```
if(x>y){
```

```
    res=z;
```

```
}
```

```
else{
```

```
    res=y-x;
```

```
}
```

```
return(res);
```

```
}
```

```
int funcao2(int x){
```

```
int res;
```

```
if(x<0){
```

```
    res=1;
```

```
}
```

```
else{
```

```
    res=res+x;
```

```
}
```

```
return(res);
```

```
}
```

Obrigatório mostrar cálculos intermédios!

Resolução pergunta 2 Grupo II

9	W	E	2	T	Y	I
5	3	4	8	7	-1	11
<del>2</del>			3	<del>1</del>	2	1

$$W = 5 + 3 = 8$$

$$T = (8 + 4) - 5 = 12 - 5 = 7$$

$$Y = 7 - 8 = -1$$

else if

$$W = 3$$

$$T = 4$$

$$Y = 2$$

I	9	7	5	3	1
11	Entrar	Entrar	Entrar	Entrar	X

Faz do else e o  
ultimo da variavel

9	X	4	2	W
<del>1</del>		3	<del>4</del>	<del>9</del>

~~IRE~~

Passa

X Y Z

Y-X

$$res 9 = 3 - 1 = 2$$

$$Y = 2 \times 2 = 4$$

T =

X	W
4	4
	T = 4

9 Passa  
ax

$$res T = 4$$

Printf 9 = 2 Printf W = 3 Printf E = 4 Printf B = 3  
Printf I = 4 Printf Y = 4 Printf I = 1

### Grupo III

1. Crie uma função que receba um número e devolva o <sup>CUBO</sup> triplo desse número. Para calcular o triplo de um número deve-se multiplicar o número recebido, três vezes.

Exemplo:

Nº inserido: 3

Resultado: 27

$$3 \times 3 \times 3$$

$$5 \times 5 \times 5$$

2. Crie uma função que receba um número e calcule a diferença entre o triplo do número seguinte e o dobro do número anterior.

Exemplo:

Nº inserido: 4

Resultado: 16

$$2 \times 4$$

$$3 \times N$$

3. Crie uma função que receba um valor que corresponde ao número de horas que um determinado funcionário trabalhou e a função deverá retornar qual o valor a receber por mês sabendo que:

- O valor/hora é de 10 euros.
- Se o número de horas for superior a 50 horas, deverá somar ao valor final 95 euros correspondentes ao subsídio de alimentação. Caso contrário não recebe qualquer compensação extra referente ao subsídio de alimentação.
- Não existem descontos se o número de horas for inferior a 100 horas. Caso seja superior, ao valor total a receber é retirado 10% desse valor.

Exemplos:

Nº inserido: 65 (horas)

Resultado: 745 (euros)

Nº inserido: 110 (horas)

Resultado: 1075 (euros) inclui corte de 10% sobre o valor total que iria receber (salário + subsídio de alimentação).

1076

4. Crie uma função que receba um valor e devolva um resultado consoante das seguintes restrições:

- Caso o valor inserido seja inferior a 4, o resultado será 0.
- Caso o valor inserido seja 4, o resultado será 1.
- Caso seja qualquer outro valor superior a 4 e inferior ou igual a 10, deverá então verificar quais os quatro valores anteriores ao número inserido e calcular a soma desses valores.
- Caso o valor inserido seja superior a 10, deverá realizar a multiplicação entre o 1º, 3º e 5º valor anterior ao número inserido.

Exemplos:

Nº inserido: 10

Resultado:  $9+8+7+6=30$

Nº inserido: 12

Resultado:  $11*9*7=693$

*Trigo Pente*

5. Realize a integração com a função *main* para que seja possível testar as suas funções de acordo com as seguintes condições:
- Solicitar ao utilizador a inserção um número e mostrar o resultado das funções criadas no ponto 1 e 2 deste grupo III.
  - Solicitar ao utilizador para introduzir um valor em horas e mostrar o resultado da função criadas no ponto 3 deste grupo III.
  - Solicitar ao utilizador a inserção de um valor para testar a função criada no ponto 4 do grupo III mostrando o resultado dessa função.

Execute todas as acções de Input/Output (scanf e printf) pela ordem do enunciado.