

Paulo Almeida

Ficha de Avaliação Sumativa

EFA Tecnológico - Técnico de Informática e Sistemas

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

Duração: 180 minutos

Nome Formando: *Paulo Almeida* Data: 10/12/2014 Formador: João Leitão

Classificação: *307 16,7 VACANCIA*

Notas Gerais:

Esta prova de natureza teórico-prática divide-se em três grupos e tem a duração de 180 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.		3	6
	2.		3	
Grupo III	1.		1	12
	2.		3	
	3.		3	
	4.		1	
	5.		2	
	6.		2	
				20

Grupo I

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:

Uma função void caracteriza-se pela ausência de tipo de dados dos argumentos.

Falso. Uma função void caracteriza-se por não devolver nada para a função main.

Na chamada de uma função é sempre necessário colocar o nome dos argumentos.

Verdadeiro

Na chamada de uma função é sempre necessário colocar o tipo de dados da função

Falso. Na chamada de uma função é necessário colocar o nome da função e o nome dos argumentos.

Na declaração de uma função é obrigatório colocar o tipo de dados de retorno da função

Verdadeiro

Paulo Almeida

Grupo II

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

```
int main(){
int x;
int y;
int z;
x=0;
y=x;
z=y;

if(x==0 && z<0){
x=1;
y=x-y*2;
}
else if(x==0 || z>10){
y=2;
x=2;
z=x+(y*2);
}
else{
z=2;
x=x+z-y;
z=z+x-y;
}

x=f1(z,y,x);

printf("%d\n",x);
printf("%d\n",y);
printf("%d\n",z);
return(0);
}

int f1(int a, int b, int c){
int res;
if(a>b){
res=a;
}
else{
res=b;
}
res=c;
return(res);
}
```

Handwritten analysis of the code execution:

- Initial state: $x=0, y=0, z=0$
- Condition $x==0 \ \&\& \ z<0$ is true. Execution enters the first `if` block:
 - $x=1$
 - $y=x-y*2 = 1-0=1$
- Condition $x==0 \ || \ z>10$ is false.
- Condition `else` is true. Execution enters the `else` block:
 - $z=2$
 - $x=x+z-y = 1+2-1=2$
 - $z=z+x-y = 2+2-1=3$
- Function call: $x = f1(z, y, x) = f1(3, 1, 2)$
- Function `f1` execution:
 - Parameters: $a=3, b=1, c=2$
 - Condition $a > b$ is true, so $res = a = 3$.
 - Final assignment: $x = res = 3$
- Final state: $x=3, y=1, z=3$

Paulo Formosa

2. Indique qual o resultado final das variáveis: x, y, z, res e i após a execução do seguinte fragmento de código:

```
#include<stdio.h>
int main(){
int x, y, z, res, i;

x=0;
y=x+1;
z=y+1;
res=1;

if(res<z){
x=f1(res,y);
z=z+1;
y=f2(z);
}
else if(res>2){
x=f1(res,z);
y=f2(x);
}
else{
x=f1(y);
res=f2(x);
z=f1(res,0);
}

for(i=0;i<5;i=i+4){
res=res+i;
}

printf("\n%d",x);
printf("\n%d",y);
printf("\n%d",z);
printf("\n%d",i);
printf("\n%d",res);

return(0);
}
```

x = 0/0 //
y = 1/2 //
z = 2/3 //
res = 1/4/8/12 //
i = 0/4/8 //

x = f1(res, y)
f1(1, 1)
res = y - x
res = 0 //

x = 0
y = 2
z = 3
res = 12
i = 8

```
int f1(int y, int x){
int res;
if(x>y){
res=0;
}
else{
res=y-x;
}
return(res);
}
```

y = f2(z)
y = f2(3)

```
int f2(int x){
int res;
do{
res=res+1;
x=x-1;
}while(x>1);
return(res);
}
```

res = 0
res = 0 + 1
res = 1
x = x - 1
x = 3 - 1
x = 2
res = 1 + 1
res = 2 //
x = 2 - 1
x = 1 //

for (i=0; i<5; i=i+4)

res = res + i

res = res + 0
res = 0
res = 0 + 4
res = 4

res = 4 + 4
res = 8
res = 8 + 4
res = 12 //

Obrigatório mostrar cálculos intermédios!

Paulo Almeida

Grupo III

1. Crie uma função que recebe um valor a pagar, um valor entregue e devolve o troco.

Exemplo:
Valor a pagar: 17,50
Valor entregue: 20
Troco: 2,50



2. Crie uma função que receba 9 números e calcule a soma dos três maiores. De notar que todos os valores devem ser positivos enviando o código de erros -99 caso algum valor seja negativo.

Exemplo:
1
9
5
7
3
6
4
8
2
Resultado: 24



3. Crie uma função que imprima os desdobramentos de números utilizando o número limite a desdobrar (N) e o nível de desdobramento (L).

Exemplo:
Insira o número limite a desdobrar: 3
Insira o nível de desdobramento: 2
Resultado
1 1
1 2
2 1
2 2
3 1
3 2



4. Crie uma função que imprima qual o troco a entregar a um determinado cliente. Para isso, a função recebe um valor a pagar e o dinheiro entregue pelo cliente e o programa calcula e mostra todas as notas e moedas (do euro) a entregar como troco ao cliente.

Assuma que sempre que for possível entrega o máximo de notas/moedas da nota/moeda mais alta e só depois passa à próxima. (ver exemplo)
Insira valor a pagar: 10,50
Insira dinheiro entregue pelo cliente: 20
Troco a entregar:
Notas de 50 euros: 0
Notas de 20 euros: 0
Notas de 10 euros: 0
Notas de 5 euros: 1
Moedas de 2 euros: 2
Moedas de 1 euro: 0
Moedas de 50 centimos: 1
Considere apenas as notas de 50,20,10 e 5 euros e as moedas de 2€, 1€ e 0.50€.
Teste o seu programa apenas com exemplos em que para o troco são somente necessárias as notas/moedas supracitadas.

Paulo
Ferreira

5. Realize a integração com a função *main* para que seja possível testar as suas funções de acordo com as seguintes condições:
 - a. Criar um menu para testar cada função criada em que a opção para testar a função corresponde ao número da alínea
 - i. 0 – Sair | 1 – Função 1 | 2 – Função 2 | etc..
 - b. Deve programar correctamente todos os resultados e enviar as mensagens para o utilizador de acordo com o resultado da função, incluindo códigos de erros detalhados.

6. Altere o *main* anterior para que sempre que a função apresente um código de erro o programa não “avance” enquanto o utilizador não inserir valores válidos para testar essa função.

Só após a escolha de um valor válido para a função e receber o resultado da mesma é que retorna ao menu de escolhas de programas (até escolher a opção para sair).