

P1

Algoritmos 2

Professor Paulo Augusto Nardi

1. Considere o código abaixo (1,5):

<pre>1 int a; 2 int b; 3 int * pa = &a; 4 int * pb = &b; 5 int * pc;</pre>	<pre>6 pc = pa; 7 pa = pb; 8 pb = pc; 9 *pc = 5; 10 *pb = 1; 11 *pa = 3;</pre>
--	--

Ao fim da execução do código acima, quais serão os valores e endereços na tabela abaixo? Considere o sistema 32 bits.

Variável	Endereço	Valor
a	0022FF20	1
b	0022FF1C	3
pa	0022FF18	0022FF1C
pb	0022FF14	0022FF20
pc	0022FF10	0022FF20

2. Altere o código abaixo para que seja realizada passagem por referência da variável mi de forma a não necessitar retorno(1,2).

<pre>int main() { float km; float mi; printf("\nDigite em km:"); scanf("%f",&km); mi=converter(km); printf("\n%.2f Km", km); printf("\n%.2f milhas", mi); return 0; }</pre>	<pre>float converter(float quilometros){ float milhas; milhas = quilometros * 0.6214f; return milhas; }</pre>
---	---

RESPOSTA NA SEGUNDA FOLHA

3. Reescreva o código abaixo de forma a substituir todas as variáveis int por ponteiros de int e alocação dinâmica (não pode haver variável do tipo int, nem mesmo os parâmetros). (3,0).

<pre>int fazMaisAlgo(int b){ b=b+5; return b; } int fazAlgo(int a){ a = a + 3; a = fazMaisAlgo(a); return a; }</pre>	<pre>int main() { int c; c=1; c=fazAlgo(c); printf("\n%d\n",c); return 0; }</pre>
---	---

RESPOSTA NA SEGUNDA FOLHA

4. Considere as atribuições abaixo:

<pre>1 int a, *b ; v 2 a=10; v 3 b=&a; v 4 *b=a; v 5 b=a; x</pre>	<pre>6 a=b; x 7 &a=b; x 8 *a=30; x 9 *b=40; v 10 40=a; x</pre>
---	--

Quais linhas possuem atribuições que deveriam ser evitadas e/ou não são permitidas? (0,5)

5- Represente a memória do código alterado no exercício 3, na tabela abaixo(1,8):

Identificador	Endereço	Valor
c	0055FF20	00998870
a	0055FF1C	00998870
b	0055FF18	00998870
<malloc>	00998870	9

P1

Algoritmos 2

Professor Paulo Augusto Nardi

<pre>2) int main() { float km; float mi; float *ponteiro1; ponteiro1 = &mi; printf("\nDigite em km:"); scanf("%f",&km); converter(km, ponteiro1); printf("\n%.2f Km", km); printf("\n%.2f milhas", mi); return 0; }</pre>	<pre>void converter(float quilometros, float *ponteiro2){ *ponteiro2 = quilometros * 0.6214f; }</pre>	<pre>3) void fazMaisAlgo(int *b){ *b=*b+5; } void fazAlgo(int *a){ *a = *a + 3; fazMaisAlgo(a); } int main() { int *c; c = (int*)malloc(sizeof(int)); *c=1; fazAlgo(c); printf("\n%d\n",*c); free(c); return 0; }</pre>
---	--	--

P1
Algoritmos 2
Professor Paulo Augusto Nardi