

EXERCÍCIOS

Para os próximos exercícios, considere que int e float ocupam 4 bytes; Double ocupa 8 bytes e char ocupa 1 byte. Ainda, suponha que o programa será compilado e executado em um sistema de 32 bits, portanto as variáveis ponteiros sempre ocupam 4 bytes (em um sistema 64 bits, os ponteiros ocupam 8 bytes). Considere, também, que a alocação de memória é feita sequencialmente, em ordem decrescente.

- 1- Preencha a tabela abaixo, conforme o programa for executado e escreva quais saídas serão apresentadas na tela. Considere que a primeira variável será endereçada em 0055FF10.

```
float a,b;

float *c,*d,*e;
a=40;
b=20;
c=&a;
d=&b;
printf("\n%p%f",&a,a);
printf("\n%p%f",&b,b);
printf("\n%p %p %f",&c,c,*c);
printf("\n%p%p %f",&d,d,*d);
a = *c + *d + 1;
e=c;
c=d;
d=e;
*c = *d + *c + 2;
printf("\n%p%f",&a,a);
printf("\n%p%f",&b,b);
printf("\n%p %p %f",&c,c,*c);
printf("\n%p %p %f",&d,d,*d);
```

Identificador	Endereço	Valor
a	0055FF10	40-61
b	0055FF14	20-83
c	0055FF18	0055FF10 0055FF14
d	0055FF1C	0055FF14 0055FF10
e	0055FF20	0055FF10

- 0060FEF8 | 40.000000
- 0060FEF4 | 20.000000
- 0060FEF0 | 0060FEF8 | 40.000000
- 0060FEEC | 0060FEF4 | 20.000000
- 0060FEF8 | 61.000000
- 0060FEF4 | 83.000000
- 0060FEF0 | 0060FEF4 | 83.000000
- 0060FEEC | 0060FEF8 | 61.000000

- 2- Preencha a tabela da próxima página, conforme o programa for executado e escreva quais saídas serão apresentadas na tela. Considere que a primeira variável será endereçada em 005F0010.

```
float a,b;
float *c,*d,*e;
double f,*g;
a=10;
b=20;
c=&a;
d=&b;
g=&f;
*g=10;
printf("\n%p %f",&a,a);
printf("\n%p %f",&b,b);
printf("\n%p %p %f",&c,c,*c);
printf("\n%p %p %f",&d,d,*d);
printf("\n%p %p %f",&g,g,*g);
e=c;
c=d;
d=e;
*g = *c + *d + *e + a + f+1;
printf("\n%p %f",&a,a);
printf("\n%p %f",&b,b);
printf("\n%p %p %f",&c,c,*c);
printf("\n%p %p %f",&d,d,*d);
printf("\n%p %p %f",&g,g,*g + 1);
```

Identificador	Endereço	Valor
a	005F0010	10
b	005F0014	20
c	005F0018	005F0010 005F0014
d	005F001C	005F0014 005F0010
e	005F0020	005F0010
f	005F0028	40-61
g	005F0030	005F0028

0060FEF8 10.000000

0060FEF4 20.000000

0060FEF0 0060FEF8 10.000000

0060FEEC 0060FEF4 20.000000

0060FEDC 0060FEE0 10.000000

0060FEF8 10.000000

0060FEF4 20.000000

0060FEF0 0060FEF4 20.000000

0060FEEC 0060FEF8 10.000000

0060FEDC 0060FEE0 62.000000

3- Preencha a tabela da próxima página, conforme o programa for executado e escreva quais saídas serão apresentadas na tela. Considere que a primeira variável será endereçada em 00552100.

Float x,d;

```
float *v,*w,*t;
x=10;
d=20;
w=&x;
t=&d;
printf("\n%p%f",&x,x);
printf("\n%p%f",&d,d);
printf("\n%p %p %f",&w,w,*w);
printf("\n%p %p %f",&t,t,*t);
d = *t + x;
v=w;
w=t;
t=v;
*t = *v + 5;
printf("\n%p%f",&x,x);
printf("\n%p%f",&d,d);
printf("\n%p %p %f",&w,w,*w);
printf("\n%p %p %f",&t,t,*t);
```

Identificador	Endereço	Valor
x	00552100	10 -15
d	00552104	20 -30
v	00552108	00552100
w	0055211C	00552100 00552104
t	00552120	00552104 00552100

0060FEF8 10.000000

0060FEF4 20.000000

0060FEF0 0060FEF8 10.000000

0060FEEC 0060FEF4 20.000000

0060FEF8 15.000000

0060FEF4 30.000000

0060FEF0 0060FEF4 30.000000

0060FEEC 0060FEF8 15.000000

4- Considere o trecho de código abaixo:

```
float a,b;  
float *c, *d,*e;  
double f;  
double *g;  
a=1;b=2;f=3;
```

VARIAVEL	ENDEREÇO	VALOR
a	0055FF10	1
b	0055FF14	2
*c	0055FF18	
*d	0055FF1C	
*e	0055FF20	
f	0055FF28	3
*g	0055FF30	

Identifique quais atribuições abaixo não são recomendadas (ou mesmo, que não podem ser realizadas).

f=10; Pode ser realizada

a=b; Pode ser realizada

a=&b; Não pode ser realizada

a=*b; Não pode ser realizada

a=c; Não pode ser realizada

a=&c; Não pode ser realizada

a=*c; Pode ser realizada

a=g; Não pode ser realizada

&a=c; Não pode ser realizada

c=a; Não pode ser realizada

c=&a; Pode ser realizada

c=*a; Não pode ser realizada.

c=d; Pode ser realizada porém não é aconselhado, pois não há nada em d

c=&d; Pode ser realizada

c=*d; Não pode ser realizada

c=g; Pode ser realizada

*c=a; Pode ser realizada

*c=&a; Pode ser realizada

*c=*a; Não pode ser realizada

*c=10; Pode ser realizada

*c=d; Não pode ser realizada

*c=*d; Não pode ser realizada

*c=&d; Pode ser realizado caso c aponte para um ponteiro

*c=*g; Pode ser realizado porem c não aponta para nenhuma variável

*g=*c; Pode ser realizado

g=c; Pode ser realizado

Obs: caso não saiba a resposta, tente usar a tabela para encontrar a explicação. Lembre-se de perguntar: o valor a ser atribuído é do mesmo tipo? Se a atribuição for feita a uma variável ponteiro, então você deve fazer duas perguntas: o valor atribuído é um endereço de memória? Este endereço de memória permite armazenar valores do tipo declarado no ponteiro?

Obs2: caso não consiga responder, teste no DevC ou CodeBlock para encontrar a resposta. Uma vez que obtenha a resposta, tente achar a explicação. Mas não se esqueça de anotar qual atribuição você teve dificuldade (e precisou da ajuda do DevC ou Codeblock) e envie email para pedir a explicação e confirmar sua resposta.