

Exercícios

Para os próximos exercícios, considere que int e float ocupam 4 bytes; Double ocupa 8 bytes e char ocupa 1 byte. Considere, também, que a alocação de memória é feita sequencialmente, em ordem decrescente.

1- Dado o trecho de código abaixo:

```
int a;  
int *b;  
a=30;  
b=&a;
```

Preencha a tabela de forma que ela represente a memória ao fim da execução.

Identificador	Endereço	Valor
a	005FF28	30
b	005FF2C	005FF28

2- Dado o trecho de código abaixo:

```
float x;  
float *y;  
x=3.2f;  
y=&x;
```

Preencha a tabela de forma que ela represente a memória ao fim da execução.

Identificador	Endereço	Valor
x	005FF14	3.2f
y	005FF18	005FF14

3- Dado o trecho de código abaixo:

```
int a;  
int *b;  
a=4;  
b=&a;  
printf("\n %p %d",&a,a);  
printf("\n %p %p %d", &b,b,*b);
```

Preencha a tabela de forma que ela represente a memória ao fim da execução.
O que será apresentado nos dois printf?

Identificador	Endereço	Valor
a	005FF10	4
b	005FF14	005FF10

No primeiro “printf” será apresentado o endereço de “a” e o que está armazenado nele e o segundo “printf” irá apresentar, em ordem, o endereço de “b”, o valor armazenado na variável “b” e o valor armazenado no endereço que “b” aponta.

- 4- Como faço para apresentar o valor de uma variável qualquer em um printf? Dê um exemplo com uma variável do tipo int.

```
int a;  
a = 5;  
printf(“%d”, a);
```

- 5- Como faço para apresentar o endereço de uma variável qualquer em um printf? Dê um exemplo com uma variável do tipo int.

```
int a;  
a = 5;  
printf(“%d”, &a);
```

- 6- Agora dê um exemplo de como apresentar o valor de uma variável do tipo “ponteiro” de int.

```
int a;  
int *b  
a = 5;  
b = &a;  
printf(“%p”, b);
```

- 7- Agora dê um exemplo de como apresentar o endereço de uma variável do tipo “ponteiro” de int.

```
int a;  
int *b  
a = 5;  
b = &a;  
printf(“%p”, &b);
```

- 8- Agora dê um exemplo de como apresentar o valor que está no espaço de memória armazenado por um “ponteiro” de int.

```
int a;  
  
int *b  
  
a = 5;  
  
b = &a;  
  
printf(“%p”, *b);
```