

Exercícios Unidade00m:

1)

- Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;  
int * y = & x;  
printf("\n%i", x);  
printf("\n%p", &x);  
printf("\n%p", y);  
printf("\n%p", &y);  
printf("\n%i", *y);
```

Memória		Tela
	75h	
	...	
	CDh	

Resposta:

Memoria	
x = 10	10h
Y = 10h	11h
	12h
	13h
	14h

Tela
10
10h
10h
11h
10

2)

```
int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);
```

Respotas:

Memoria	
X1 = lixo => 11 => 33	10h
X2 = lixo => 22 => 11	11h
X3 = lixo => 33 => 0	12h
P => 10h => 12h	13h

Tela
Cont: 33 11 0 0
Addr: 10h 11h 12h 13h

3)

- Faça o quadro de memória e mostre a saída na tela:

```
int *x1;      int x2;      int *x3;

x1 = (int *) malloc (sizeof(int));
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);

*x1 = 20;
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);

x2 = *x1;
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);

*x3 = x2 * *x1;
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);

x3 = &x2;
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);

x2 = 15;
printf("\nx1(%p)(%i)(%p) x2(%i)(%p) x3(%p)(%i)(%p)", x1, *x1, &x1, x2, &x2, x3, *x3, &x3);
```

Resposta:

Memoria	
lixo => 20	10h
Lixo	11h
Lixo	12h
Lixo	13h
X1 = 10h	14h
X2 = lixo => 20 => 15	15h
X3 = 18h	16h
	17h
lixo => 400 => 15h	18h

Tela
X1(14h)(lixo)(10h)
X2(lixo)(15h)
X3(18h)(lixo)(16h)
X1(14h)(20)(10h)
X2(lixo)(15h)
X3(18h)(lixo)(16h)
X1(14h)(20)(10h)
X2(20)(15h)
X3(18h)(lixo)(16h)
X1(14h)(20)(10h)
X2(20)(15h)
X3(18h)(400)(16h)
X1(14h)(20)(10h)
X2(20)(15h)
X3(18h)(20)(16h)
X1(14h)(20)(10h)

X2(15)(15h)
X3(18h)(15)(16h)

4)

- Mostre a saída na tela

<pre>double a; double *p, *q; a = 3.14; printf("%f\n", a); p = &a; *p = 2.718; printf("%f\n", a); a = 5; printf("%f\n", *p);</pre>	<pre>p = NULL; p = (double*) malloc(sizeof(double)); *p = 20; q = p; printf("%f\n", *p); printf("%f\n", a); free(p); printf("%f\n", *q);</pre>
--	--

Resposta:

Saida
3.14
2.718
5.0
20.0
5.0
0.0