

Aula 02 – Exercícios

2.1. Quando desenvolvemos um programa para computadores pessoais há restrições de acesso à determinados endereços de memória. Por quê? Quais endereços um programa tem acesso?

2.2. O que é um apontador (ou ponteiro) em linguagens de programação?

2.3. Considere as seguintes variáveis e atribuições:

```
int *p, x;
p = &x;
x = 10;
```

Qual a relação que se pode estabelecer entre **p**, ***p**, **&x** e **x**?

2.4. (a) Analise o programa abaixo e preencha os valores das variáveis na memória (representada pela tabela Memória ao lado do programa) quando a execução estiver na **linha 11**.

(b) O que será exibido na tela?

```
01  #include <stdio.h>
02
03  int main() {
04      int *c, v;
05
06      c = &v;
07      v = 25;
08      *c *= 4;
09
10      printf( "%d\n", v );
11
12      return 0;
13  }
```

Memória

Endereço (base decimal)	Nome da Variável	Tipo da Variável	Conteúdo (base decimal)
2258544	C		
2258548	V		

2.5. (a) Analise o programa abaixo e preencha os valores das variáveis na memória (representada pela tabela Memória ao lado do programa) quando a execução estiver na **linha 14**.

(b) O que será exibido na tela?

```
01  #include <stdio.h>
02
03  int main() {
04      int a;
05      char *pt, ch1, ch2;
06      double v = 4.0;
07
08      a = 5;
09      ch1 = 'R';
10      pt = &ch1;
11      ch2 = *pt;
12      printf( "%f", v + 8.3 );
13      v = (4*a) + ((double)ch1);
14
15      return 0;
16  }
```

Memória

Endereço (base decimal)	Conteúdo (base decimal)	Nome da Variável	Tipo da Variável
1326864			
1326868			
1326872			
1326876			
1326880			

2.6. O que significa um apontador apontar para **NULL**?

Exemplo: `int *p=NULL;`

2.7. Faça um programa no qual são definidas duas variáveis (x e y) do tipo *float*, com valor inicial 0.0 (zero). Deve-se exibir na tela os endereços e os conteúdos dessas variáveis. O usuário deverá ser questionado para digitar um dos dois endereços e um novo valor. O programa deverá atribuir esse valor ao conteúdo do endereço de memória digitado pelo usuário. Os endereços e os conteúdos das variáveis deverão ser exibidos novamente, assim como o questionamento para digitar um dos dois endereços e um novo valor. Esse processo deverá se repetir até que um endereço diferente dos endereços das variáveis x e y seja digitado pelo usuário. Nesse caso o programa termina.

2.8. Considere o programa abaixo:

```

01  #include <stdio.h>
02  int main() {
03      int v, *p, aux;
04      v = 250;
05      printf( "[v] end: %d cont: %d\n", &v, v );
06      printf( "[p] end: %d cont: %d pt: %d\n\n", &p, p, *p );
07      printf( "Digite um valor para p: " );
08      scanf( "%d", &p );
09      printf( "[v] end: %d cont: %d\n", &v, v );
10      printf( "[p] end: %d cont: %d pt: %d\n\n", &p, p, *p );
11      *p = 612;
12      printf( "[v] end: %d cont: %d\n", &v, v );
13      printf( "[p] end: %d cont: %d pt: %d\n\n", &p, p, *p );
14      printf( "Digite um valor para v: " );
15      scanf( "%d", &v );
16      printf( "[v] end: %d cont: %d\n", &v, v );
17      printf( "[p] end: %d cont: %d pt: %d\n\n", &p, p, *p );
18      printf( "Digite um valor para v: " );
19      scanf( "%d", &aux );
20      *p = aux;
21      printf( "[v] end: %d cont: %d\n", &v, v );
22      printf( "[p] end: %d cont: %d pt: %d\n\n", &p, p, *p );
23      return 0;
24  }

```

- a) Qual o único valor possível de ser digitado pelo usuário no *scanf* da linha 08? Por quê?
- b) Qual a sequência de valores atribuídas à variável *v* no decorrer da execução do programa? Indique em quais linhas essas atribuições são realizadas e se elas são realizadas de forma direta ou indireta. No caso de atribuições indiretas, indique como a atribuição é feita.