

UFRN - IMD - LP1

Lista de exercícios (Aula03)

1. A chamada de função `misterio(&i, &j)` tem um problema. Qual é? Antes da chamada, temos a seguinte linha de comando: `int i=6, j=10;`

```
void misterio(int *p, int *q){  
    int *temp;  
    *temp = *p;  
    *p = *q;  
    *q = *temp;  
}
```

2. O que as linhas abaixo fazem? Dica: Use o gdb para confirmar sua resposta.

```
int i=99, j;  
int *p;  
p = &i;  
j = *p + 100;
```

3. O que as linhas abaixo fazem? Dica: Use o gdb para confirmar sua resposta.

```
int a=5, b=12;  
int *p;  
int *q;  
p = &a;  
q = &b;  
int c = *p + *q;
```

4. O que as linhas abaixo fazem? Dica: Use o gdb para confirmar sua resposta.

```
int i=7, j=3;  
int *p;  
int **r;  
p = &i;  
r = &p;  
int c = **r + j;
```

5. Escreva uma função recursiva para determinar o menor elemento de um vetor não-ordenado de inteiros.

6. (mostrar) Dado o programa abaixo, completar as tabelas 1 e 2 manualmente, e em seguida, utilizando o gdb:

```

main() {
    int i, j, *p_1, *p_2, **p_p_1, **p_p_2;
    i = 4;  j = 5;
    p_1 = &i;
    p_2 = &j;
    p_p_1 = &p_2;
    p_p_2 = &p_1;
}

```

Tabela 1:

Variável	i	j	p_1	p_2	p_p_1	p_p_2
Conteúdo	4	5				
Endereço	1000	1007	1030	1053	1071	1079
End. GDB						

Tabela2:

Expressão	i	*p_2	&i	&p_2	**p_p_1	*p_p_2	&*p_1	j	*p_1	*&p_1
Resultado										

7. Dado o programa abaixo, completar as tabelas 3 e 4 manualmente, e em seguida, utilizando o gdb:

```

main() {
    int i, j, *p_1, *p_2, **p_3, ***p_4;
    i = 4;
    j = 5;
    p_1 = &j;
    p_2 = &i;
    p_3 = &p_1;
    p_4 = &p_3;
}

```

Tabela 3:

Variável	i	j	p_1	p_2	p_3	p_4
Conteúdo	4	5				

Endereço	1000	1007	1030	1053	1071	1079
End. GDB						

Tabela 4:

Expressão	i	*p_2	&j	&p_2	**p_3	**p_4	***p_4	*p_1	*&p_2	*p_4
Resultado										

8. Dado o Programa abaixo, completar as Tabelas 5 e 6 manualmente, e em seguida, utilizando o gdb:

```
main()
{
    int i, j, *p_1, **p_2, **p_3, ***p_4;
    i = 4;
    j = 5;
    p_1 = &j;
    p_2 = &p_1;
    p_3 = p_2;
    p_4 = &p_3;
}
```

Tabela 5:

Variável	i	j	p_1	p_2	p_3	p_4
Conteúdo	4	5				
Endereço	1000	1007	1030	1053	1071	1079
End. GDB						

Tabela 6:

Expressão	i	*p_2	&j	&p_2	**p_3	**p_4	***p_4	*p_1	*&p_2	*p_4
Resultado										

9. Dado o Programa abaixo, completar as Tabelas 7 e 8 manualmente, e em seguida, usando o gdb:

```
main() {
    int i, *p_1, *p_2, v[10];
    p_1 = v;
    p_2 = p_1;
    for (i = 0; i < 10; i++)
        { v[i] = (2*i)+1; p_2++; }
    for (i = 0; i < 10; i++)
        { p_2--; std::cout << " [" << *p_2 << "]" " ; }
}
```

Tabela 7:

[illegible]

Tabela 8 (Valores impressos com: `std::cout << " [" << *p_2 << "]" " ;`).

[illegible]