Apêndice - Códigos

1 Alocação estática

1. **Acesso direto** (sem referências):

   – *Variáveis escalares:* `http://cpp.sh/33n3i`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int var1;
    cout << var1 << endl;
    var1 = 2;
    cout << var1 << endl;
    return 1;
}
```

Out:

```
12312309
2
```

   – *Vetores:* `http://cpp.sh/2fkt`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int v1[3];
    cout << v1[0] << endl;
    v1[0] = 5;
    cout << v1[0] << endl;
    return 1;
}
```

Out:

```
42834090
5
```

2. **Acesso direto** (com referências):

   – *No mesmo escopo:* `http://cpp.sh/5owne`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int var1;
    int & var2 = var1;
    cout << var1 << endl;
    var2 = 2;
    cout << var1 << endl;
    return 1;
}
```

Out:

```
984203984
2
```

   – *Como parâmetro:* `http://cpp.sh/7rkej`

```
#include <iostream>
using namespace std;
void inc (int & x) {
    x = x + 1;
}
int main(int argc, char * argv[]) {
    int var1 = 2;
    cout << var1 << endl;
    inc(var1);
    cout << var1 << endl;
    return 1;
}
```

Out:

```
2
3
```

3. **Acesso indireto** (com ponteiros, C++):

   – *No mesmo escopo, com variáveis escalares:* `http://cpp.sh/7phkr`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int var1;
    int * var2 = &var1;
    cout << var1 << ";" << *var2 << endl;
    *var2 = 5;
    cout << var1  << ";" << *var2 << endl;
    return 1;
}
```

Out:

```
128372103, 128372103
5;5
```

– *No mesmo escopo, com vetores:* http://cpp.sh/8opxh

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int v1[3];
    int * v2 = v1;
    cout << v1[0] << ";" << v2[0] << endl;
    v2[0] = 5;
    cout << v1[0] << ";" << v2[0] << endl;
    return 1;
}
```

Out:

```
9621827, 9621827
5;5
```

– *Como parâmetro, com variável escalar:* http://cpp.sh/8eqp3

```
#include <iostream>
using namespace std;
void inc (int * x) {
    *x = *x + 1;
}
int main(int argc, char * argv[]) {
    int var1 = 2;
    cout << var1 << endl;
    inc(&var1);
    cout << var1 << endl;
    return 1;
}
```

Out:

```
2
3
```

– *Como parâmetro, com vetores:* http://cpp.sh/4daq5

```
#include <iostream>
using namespace std;
void zerar (int * x, int i) {
    x[i] = 0;
}
int main(int argc, char * argv[]) {
    int v1[3] = {1, 0, 1};
    cout << v1[2] << endl;
    zerar(v1, 2);
    cout << v1[2] << endl;
    return 1;
}
```

Out:

```
1
0
```

2 Alocação dinâmica

1. **Acesso indireto** (com ponteiros):

   – *No mesmo escopo, com variáveis escalares:* `http://cpp.sh/85yq3`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int * var1 = new int;
    cout << *var1 << endl;
    *var1 = 5;
    cout << *var1 << endl;
    delete var1;
    return 1;
}
```

   Out:

```
1283798
5
```

   – *No mesmo escopo, com vetores:* `http://cpp.sh/2c7do`

```
#include <iostream>
using namespace std;
int main(int argc, char * argv[]) {
    int * v1 = new int[3];
    cout << v1[0] << endl;
    v1[0] = 5;
    cout << v1[0] << endl;
    return 1;
}
```

   Out:

```
8723498
5
```

   – *Como parâmetro, com variável escalar:* `http://cpp.sh/5awj`

```
#include <iostream>
using namespace std;
void inc (int * x) {
    *x = *x + 1;
}
int main(int argc, char * argv[]) {
    int * var1 = new int;
    *var1 = 2;
    cout << *var1 << endl;
    inc(var1);
    cout << *var1 << endl;
    return 1;
}
```

   Out:

```
2
3
```

– *Como parâmetro, com vetores:* http://cpp.sh/9isn2

```
#include <iostream>
using namespace std;
void zerar (int * x, int i) {
    x[i] = 0;
}
int main(int argc, char * argv[]) {
    int * v1 = new int[3];
    v1[2] = 3;
    cout << v1[2] << endl;
    zerar(v1,2);
    cout << v1[2] << endl;
    return 1;
}
```

Out:

```
3
0
```

2. **Acesso misto**:

– *No mesmo escopo:* http://cpp.sh/9yqno

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
  int n;
  cin >> n;
  int v1[n];
  cout << v1[2] << endl;
  v1[2] = 2;
  cout << v1[2] << endl;
}
```

Out:

```
28732731
2
```

– *Como parâmetro:* http://cpp.sh/473qm

```
#include <iostream>
#include <cstdlib>
using namespace std;
void zerar(int * x, int i) {
    x[i] = 0;
}
int main()
{
  int n;
  cin >> n;
  int v1[n];
  v1[2] = 2;
  cout << v1[2] << endl;
  zerar(v1, 2);
  cout << v1[2] << endl;
}
```

Out:

```
2
0
```