**Exercícios aplicando Vetores**

**Soluções**

1. Faça **teste de mesa** para os seguintes programas aplicando vetores e mostre conteúdo final dos vetores definidos:

void main ( ) {

static int vet [ ] = { 10, 20, 35, 40, 50 };

int c, aux;

for ( c = 0; c < 4; c++ )

if ( vet [ c ] == 20 )

vet [ c ] = 5;

else if ( vet [ c ] % 2 == 1 )

aux = vet [ c ] / 10;

else if ( vet [ c ] >= 40 )

vet [ c ] = vet [ c-1 ];

vet [ 0 ] = vet [ aux ];

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iter** | **c** | **c < 4** | **vet[ ]** | **aux** |
| - | - | - | { 10, 20, 35, 40, 50 } | - |
| 1 | 0 | V | { 10, 20, 35, 40, 50 } | - |
| 2 | 1 | V | { 10, **5**, 35, 40, 50 } | - |
| 3 | 2 | V | { 10, 5, 35, 40, 50 } | **3** |
| 4 | 3 | V | { 10, 5, 35, **35**, 50 } | 3 |
| - | 4 | F | { **35**, 5, 35, 35, 50 } | 3 |

void main ( ) {

static int vet [ 9 ];

int ind, aux = 3;

for ( ind = 0; ind < 9; ind++ )

vet [ ind ] = ind + 1;

for ( ind = 2; ind < 7; ind++ )

if ( vet [ ind ] >= 1 && vet [ ind ] <= 3 )

vet [ ind ] = ind + 10;

else if ( vet [ ind ] % 2 == 0 )

vet [ ind ] = vet [ ind ] + vet [ ind + 3 ];

else

aux = vet [ ind ];

vet [ aux ] = vet [ 8 ] \* 2;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iter** | **ind** | **ind < 7** | **vet[ 9 ]** | **aux** | **Observação** |
| - | - | - | { 0, 0, 0, 0, 0, 0, 0, 0, 0 } | 3 |  |
| - | 0 - 8 | - | { 1, 2, 3, 4, 5, 6, 7, 8, 9 } | 3 | 1º *for* |
| 1 | 2 | V | { 1, 2, **12**, 4, 5, 6, 7, 8, 9 } | 3 | Início 2º *for* |
| 2 | 3 | V | { 1, 2, 12, **11**, 5, 6, 7, 8, 9 } | 3 |  |
| 3 | 4 | V | { 1, 2, 12, 11, 5, 6, 7, 8, 9 } | **5** |  |
| 4 | 5 | V | { 1, 2, 12, 11, 5, **15**, 7, 8, 9 } | 5 |  |
| 5 | 6 | V | { 1, 2, 12, 11, 5, 15, 7, 8, 9 } | **7** |  |
| - | 7 | F | { 1, 2, 12, 11, 5, 15, 7, **18**, 9 } | 7 |  |

void main ( ) {

static int vetA [ 5 ] = { 4, 7, 1, 20, 87 };

static int vetB [ 5 ] = { 1, 5, 8, 3, 6 };

static int vetC [ 5 ];

int i;

for ( i = 0; i < 5; i++ )

if ( vetA [ i ] % 2 == 0 )

vetC [ i ] = vetA [ i ];

else if ( vetB [ i ] % 2 == 0 )

vetC [ i ] = vetB [ i ];

}

vetA = { 4, 7, 1, 20, 87 }

vetB = { 1, 5, 8, 3, 6 }

|  |  |  |  |
| --- | --- | --- | --- |
| **Iter** | **i** | **i < 5** | **vetC [ ]** |
| - | - | - | { 0, 0, 0, 0, 0 } |
| 1 | 0 | V | { **4**, 0, 0, 0, 0 } |
| 2 | 1 | V | { 4, **0**, 0, 0, 0 } |
| 3 | 2 | V | { 4, 0, **8**, 0, 0 } |
| 4 | 3 | V | { 4, 0, 8, **20**, 0 } |
| 5 | 4 | V | { 4, 0, 8, 20, **6** } |
| - | 5 | F | { 4, 0, 8, 20, 6 } |

void main ( ) {

static int vetA [ 5 ] = { 4, 7, 1, 20, 87 };

static int vetB [ 5 ] = { 1, 5, 8, 3, 6 };

static int vetC [ 5 ], vetD [ 5 ];

int i, j;

for ( j = 0, i = 4; i > 0; )

vetC [ j++ ] = vetA [ i-- ];

for ( j = 5, i = 0; i < 5; i++ )

if ( vetB [ i ] > 3 )

vetD [ --j ] = vetB [ i ];

}

vetA = { 4, 7, 1, 20, 87 }

vetB = { 1, 5, 8, 3, 6 }

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iter** | **i** | **j** | **i > 0** | **vetC [ ]** | **Observação** |
| - | - | - | - | { 0, 0, 0, 0, 0 } | 1º *for* |
| 1 | 4 | 0 | V | { **87**, 0, 0, 0, 0 } |  |
| 2 | 3 | 1 | V | { 87, **20**, 0, 0, 0 } |  |
| 3 | 2 | 2 | V | { 87, 20, **1**, 0, 0 } |  |
| 4 | 1 | 3 | V | { 87, 20, 1, **7**, 0 } |  |
| - | 0 | 4 | F |  |  |
|  |  |  |  |  |  |
| **Iter** | **i** | **j** | **i < 5** | **vetD [ ]** | **Observação** |
| - | - | - | - | { 0, 0, 0, 0, 0 } | 2º *for* |
| 1 | 0 | 5 | V | { 0, 0, 0, 0, 0 } |  |
| 2 | 1 | 4 | V | { 0, 0, 0, 0, **5** } |  |
| 3 | 2 | 3 | V | { 0, 0, 0, **8**, 5 } |  |
| 4 | 3 | 3 | V | { 0, 0, 0, 8, 5 } |  |
| 5 | 4 | 2 | V | { 0, 0, **6**, 8, 5 } |  |
| - | 5 | 2 | F |  |  |