

DESAFIO 4: Em aula

Vale até 

Programa A:

Código	Layout de memória																		
<pre>//Programa A #include <stdio.h> int f(int x, int y) { int z; x=1; y=1; z = x+y; return z; } int main(void) { int x=0,y=0,z=0; printf("%d %d %d\n",x,y,z); z = f(x,y); printf("%d %d %d\n",x,y,z); return 0; }</pre>	<table><tr><th colspan="2">Ra Main()</th></tr><tr><td>X</td><td>0</td></tr><tr><td>Y</td><td>0</td></tr><tr><td>Z</td><td>2</td></tr><tr><th colspan="2">RA f(0,0)</th></tr><tr><td>X</td><td>1</td></tr><tr><td>Y</td><td>1</td></tr><tr><td>Z</td><td>2</td></tr><tr><th colspan="2">Code</th></tr></table> <p>Saída:</p> <p>0 0 0</p> <p>0 0 2</p>	Ra Main()		X	0	Y	0	Z	2	RA f(0,0)		X	1	Y	1	Z	2	Code	
Ra Main()																			
X	0																		
Y	0																		
Z	2																		
RA f(0,0)																			
X	1																		
Y	1																		
Z	2																		
Code																			
Passagem por: Valor ou cópia																			

Código	Layout de memória																														
<pre>//Programa A_trace #include <stdio.h> int f(int x, int y) { int z; printf("Endereços (l-values) no RA de f:\n"); printf("%X %X %X\n",&x,&y,&z); x=1; y=1; z = x+y; printf("Valores (r-values) no RA de f:\n"); printf("%d %d %d\n",x,y,z); return z; } int main(void) { int x=0,y=0,z=0; printf("Endereços (l-values) no RA da main:\n"); printf("%X %X %X\n",&x,&y,&z); printf("Valores (r-values) no RA da main:\n"); printf("%d %d %d\n",x,y,z); z = f(x,y); printf("Novos valores (r-values) no RA da main após chamar f:\n"); printf("%d %d %d\n",x,y,z); return 0; }</pre>	<table><tr><th colspan="3">Ra Main(void)</th></tr><tr><td>2898D86C</td><td>x</td><td>0</td></tr><tr><td>2898D870</td><td>y</td><td>0</td></tr><tr><td>2898D874</td><td>z</td><td>2</td></tr><tr><th colspan="3">RA da f(0,0)</th></tr><tr><td>2898D83C</td><td>x</td><td>1</td></tr><tr><td>2898D838</td><td>y</td><td>1</td></tr><tr><td>2898D844</td><td>z</td><td>2</td></tr><tr><th colspan="3">Heap</th></tr><tr><th colspan="3">Code</th></tr></table> <p>Endereços (l-values) no RA da main: 2898D86C 2898D870 2898D874 Valores (r-values) no RA da main: 0 0 0 Endereços (l-values) no RA de f: 2898D83C 2898D838 2898D844 Valores (r-values) no RA de f: 1 1 2 Novos valores (r-values) no RA da main após chamar f: 0 0 2</p>	Ra Main(void)			2898D86C	x	0	2898D870	y	0	2898D874	z	2	RA da f(0,0)			2898D83C	x	1	2898D838	y	1	2898D844	z	2	Heap			Code		
Ra Main(void)																															
2898D86C	x	0																													
2898D870	y	0																													
2898D874	z	2																													
RA da f(0,0)																															
2898D83C	x	1																													
2898D838	y	1																													
2898D844	z	2																													
Heap																															
Code																															
Passagem por: valor ou cópia																															

Programa B:

Código

```
//Programa B
#include <stdio.h>

int f(int *x, int y)
{
    *x=1;
    y=1;
    return *x+y;
}

int main(void) {
    int x=0,y=0,z=0;
    printf("%d %d %d\n",x,y,z);
    z = f(&x,y);
    printf("%d %d %d\n",x,y,z);
    return 0;
}
```

Passagem por: valor (ou cópia)

Layout de memória

Ra Main(void)		
x	1	
y	0	
z	2	
Ra da f(2898D86C, 0)		
2898D86C	x	1
2898D838	y	1
Temp		2
Code		

Saída:

0 0 0
1 0 2

Programa C:

Código

```
//Programa C
#include <stdio.h>

int y = 4;

int f(int *x)
{
    *x=1;
    return *x+y;
}

int main(void) {
    int x=0,z=0,*p;
    p = &x;
    printf("%d %d %d\n",x,y,z);
    z = f(p);
    printf("%d %d %d\n",x,y,z);
    return 0;
}
```

Passagem por: valor (ou cópia)

Layout de memória

Ra Main(void)	
x	1
z	5
p	2898D86C
Ra da f (2898D86C)	
*x	1
Temp	5
Code	
y	4

Saída:

0 4 0
1 4 5

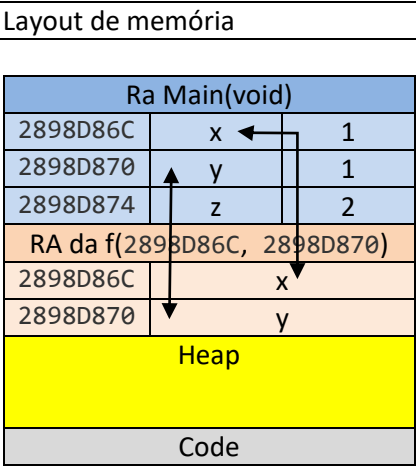
Programa D:

```
//Programa D
#include <iostream>
using namespace std;

int f(int &x, int &y)// & indica passagem por referência
{
    x=1;
    y=1;
    return x+y;
}

int main() {
    int x=0,y=0,z=0;
    cout << x << y << z << endl;//Tem a mesma função do printf
    z = f(x,y);
    cout << x << y << z << endl;
    return 0;
}
```

Passagem por: Referência



Saídas:

000
112