

Laboratório 13

Assembly: Função, Array Local e Estrutura Local

1. Traduza a função add() abaixo para Assembly:

```
struct X {
    int val;
    struct X *next;
};

int add(struct X *x) {
    int a = 0;
    for ( ; x != NULL; x = x->next)
        a += x->val;
    return a;
}
```

Utilize a função main() abaixo para testar a função add():

```
#include <stdio.h>
#define MAX 10 /* MAX deve ser maior que 1! */

struct X {
    int val;
    struct X *next;
};

int add(struct X *x);

int main(void) {
    struct X x[MAX];
    int i;

    for (i = 0; i < MAX; i++)
        x[i].val = i;

    for (i = MAX-1; i > 1; i--)
        x[i].next = &(x[i-1]);
    x[0].next = &(x[MAX-1]);
    x[1].next = NULL;
```

```
printf ("Resultado de add: %d\n", add(x));  
return 0;  
}
```

2. Traduza a função add2() abaixo para Assembly:

```
struct X {  
    int val;  
    struct X *next;  
};  
  
int add2(struct X *x) {  
    if (x == NULL) return 0;  
    else          return x->val + add2(x->next);  
}
```

Use a função main() do exercício anterior (fazendo as devidas alterações) para testar add2().

3. Traduza a função foo() abaixo para Assembly:

```
struct X {  
    int val1;  
    int val2;  
};  
  
int f(int i, int v);  
  
void foo(struct X *px, int n, int val) {  
    while (n-->0) {  
        px->val2 = f(px->val1, val);  
        px++;  
    }  
}
```

Utilize a função main() abaixo para testar a função foo():

```
#include <stdio.h>  
  
struct X {  
    int val1;  
    int val2;  
};  
  
int foo (struct X *px, int n, int val);
```

```

int f(int i, int v) {
    if (i >= v)
        return i;
    else
        return v;
}

int main (void) {
    int i;
    struct X x[3] = {{1,0}, {2,0}, {3,0}};

    printf("Antes: ");
    for (i = 0; i < 3; i++)
        printf("{%d,%d} ", x[i].val1, x[i].val2);

    foo(x, 3, 2);

    printf("\nDepois: ");
    for (i=0; i < 3; i++)
        printf("{%d,%d} ", x[i].val1, x[i].val2);
    printf("\n");

    return 0;
}

```

4. Traduza a função map2() abaixo para Assembly:

```

int f(int x);

void map2(int *um, int *outro, int n) {
    int i;
    for (i = 0; i < n; i++)
        *(outro+i) = f( *(um+i) );
}

```

Utilize a função main() abaixo para testar a função foo():

```

#include <stdio.h>
#define N 10

void map2(int *um, int *outro, int n);

int f (int x) { return x+2; }

int main (void) {
    int i;
    int a[N], b[N];
}

```

```
    for (i = 0; i < N; i++)  
        a[i] = i;  
  
    map2(a, b, N);  
  
    for (i=0;i<N;i++)  
        printf("%d\n", b[i]);  
  
    return 1;  
}
```