Exercício 1

Ex1.c

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
  }

void exibe(Lista L) {
     while( L != NULL ) {
        printf("%d\n",L->item);
        L = L->prox;
     }
  }

int main(void) {
   Lista I = no(3,no(1,no(5,NULL)));
   exibe(I);
   return 0;
}
```

```
C:\Users\P08012631\Desktop\
3
1
5

--------------------------------
Process exited after 0.02066 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 2

Ex2.c

```c
typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
  }

void exibe(Lista L) {
     printf("[");
     if(L == NULL){

     }else{
        printf("%d",L->item);
        L = L->prox;
        while( L != NULL ) {
            printf(",%d",L->item);
            L = L->prox;
        }

     }
     printf("]");
  }

int main(void) {
   Lista I =NULL;
   exibe(I);
   return 0;
}
```

```
C:\Users\P08012631\Desktop\
[]
--------------------------------
Process exited after 1.407 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Ex2.c

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
  }

void exibe(Lista L) {
     printf("[");
     if(L == NULL){

     }else{
        printf("%d",L->item);
        L = L->prox;
        while( L != NULL ) {
            printf(",%d",L->item);
            L = L->prox;
        }

     }
     printf("]");
  }

int main(void) {
   Lista I =no(3,no(1,no(5,NULL)));
```

```
C:\Users\P08012631\Desktop\
[3,1,5]
--------------------------------
Process exited after 1.469 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 3

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
 }

int tamanho(Lista L) {
    int t = 0;
    while( L ) {
       t++;
       L = L->prox;
    }
    return t;
 }

void exibe(Lista L) {
    while( L != NULL ) {
       printf("%d\n",L->item);
       L = L->prox;
    }
 }

int main(void) {
   Lista I = no(3,no(1,no(5,NULL)));
   exibe(I);
   printf("Tamanho = %d\n",tamanho(I));
```

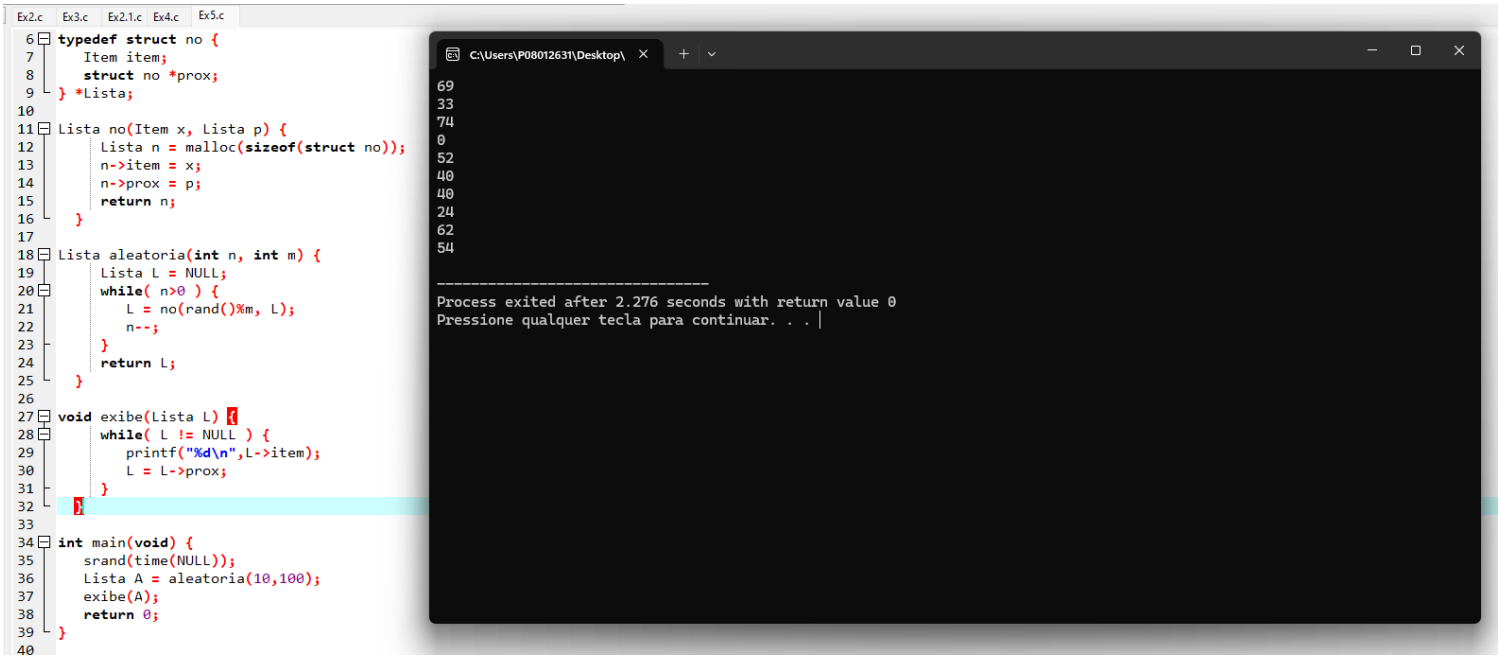
```
C:\Users\P08012631\Desktop\
3
1
5
Tamanho = 3

--------------------------------
Process exited after 0.6836 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 4

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
 }

int tamanho(Lista L) {
    int t = 0;
    while( L ) {
       t++;
       L = L->prox;
    }
    return t;
 }

int soma(Lista L) {
    int soma = 0;
    while (L != NULL) {
        soma += L->item;
        L = L->prox;
    }
    return soma;
}

void exibe(Lista L) {
```

```
C:\Users\P08012631\Desktop\
3
1
5
Tamanho = 3
A soma dos numeros acima eh: 9
--------------------------------
Process exited after 1.126 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

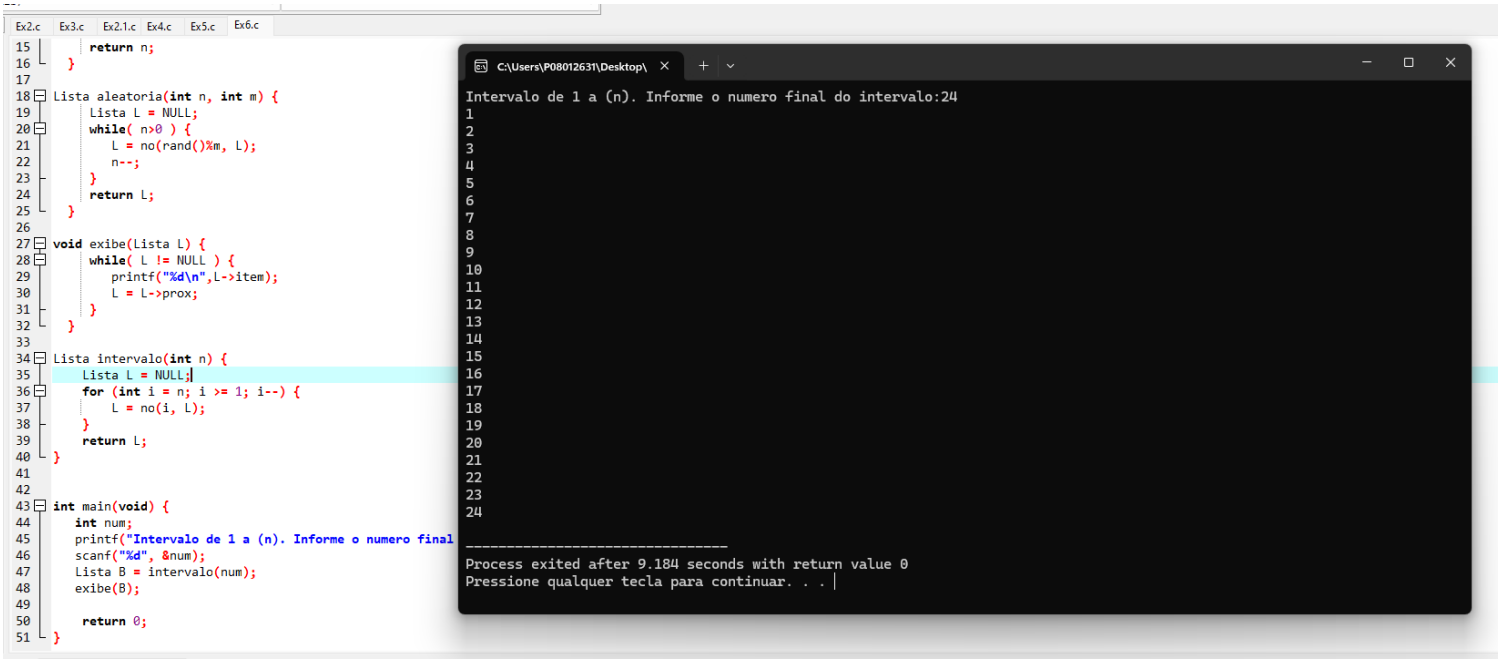
Exercício 5



```c
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
  }

Lista aleatoria(int n, int m) {
    Lista L = NULL;
    while( n>0 ) {
       L = no(rand()%m, L);
       n--;
    }
    return L;
  }

void exibe(Lista L) {
    while( L != NULL ) {
       printf("%d\n",L->item);
       L = L->prox;
    }
  }

int main(void) {
   srand(time(NULL));
   Lista A = aleatoria(10,100);
   exibe(A);
   return 0;
}
```

```
C:\Users\P08012631\Desktop\
69
33
74
0
52
40
40
24
62
54

--------------------------------
Process exited after 2.276 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 6



```c
    return n;
  }

Lista aleatoria(int n, int m) {
    Lista L = NULL;
    while( n>0 ) {
       L = no(rand()%m, L);
       n--;
    }
    return L;
  }

void exibe(Lista L) {
    while( L != NULL ) {
       printf("%d\n",L->item);
       L = L->prox;
    }
  }

Lista intervalo(int n) {
    Lista L = NULL;
    for (int i = n; i >= 1; i--) {
       L = no(i, L);
    }
    return L;
}


int main(void) {
   int num;
   printf("Intervalo de 1 a (n). Informe o numero final
   scanf("%d", &num);
   Lista B = intervalo(num);
   exibe(B);

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Intervalo de 1 a (n). Informe o numero final do intervalo:24
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

--------------------------------
Process exited after 9.184 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 7

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
  }

void anexa(Lista *A, Lista B) {
     if( !B ) return;
     while( *A )
        A = &(*A)->prox;
     *A = B;
  }

void exibe(Lista L) {
     while( L != NULL ) {
        printf("%d\n",L->item);
        L = L->prox;
     }
  }

int main(void) {
    Lista H = no(4,no(2,NULL));
    Lista I = no(3,no(1,no(5,NULL)));
    printf("H = "); exibe(H);
    printf("I = "); exibe(I);
    printf("Pressione enter");
    getchar();
```

```
C:\Users\P08012631\Desktop\
H = 4
2
I = 3
1
5
Pressione enter
```

Exercício 8

```c
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
}

Item ultimo(Lista L) {
    if (L == NULL) {
        printf("Erro fatal: a lista esta vazia.\n");
    }

    while (L->prox != NULL) {
        L = L->prox;
    }

    return L->item;
}

void exibe(Lista L) {
     while( L != NULL ) {
        printf("%d\n", L->item);
        L = L->prox;
     }
}

int main(void) {
   Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
   exibe(L);
   printf("Ultimo item da lista: %d\n", ultimo(L));

   return 0;
```

```
C:\Users\P08012631\Desktop\
3
1
5
66
6
Ultimo item da lista: 6

--------------------------------
Process exited after 0.9402 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

```c
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
}

Item ultimo(Lista L) {
    if (L == NULL) {
        printf("Erro fatal: a lista esta vazia.\n");
    }

    while (L->prox != NULL) {
        L = L->prox;
    }

    return L->item;
}

void exibe(Lista L) {
     while( L != NULL ) {
        printf("%d\n", L->item);
        L = L->prox;
     }
}

int main(void) {
   Lista L = NULL;
   exibe(L);
   printf("Ultimo item da lista: %d\n", ultimo(L));

   return 0;
}
```

```
C:\Users\P08012631\Desktop\
Erro fatal: a lista esta vazia.

--------------------------------
Process exited after 1.854 seconds with return value 3221225477
Pressione qualquer tecla para continuar. . .
```

Exercício 9

```c
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

Item maximo(Lista L) {
    if (L == NULL) {
        printf("Erro fatal: a lista esta vazia.\n");
    }

    Item max = L->item;
    while (L != NULL) {
        if (L->item > max) {
            max = L->item;
        }
        L = L->prox;
    }

    return max;
}

void exibe(Lista L) {
    while( L != NULL ) {
        printf("%d\n", L->item);
        L = L->prox;
    }
}

int main(void) {
    Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
    exibe(L);
    printf("Maior item da lista: %d\n", maximo(L));

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
3
1
5
66
6
Maior item da lista: 66

--------------------------------
Process exited after 0.7492 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

```c
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

Item maximo(Lista L) {
    if (L == NULL) {
        printf("Erro fatal: a lista esta vazia.\n");
    }

    Item max = L->item;
    while (L != NULL) {
        if (L->item > max) {
            max = L->item;
        }
        L = L->prox;
    }

    return max;
}

void exibe(Lista L) {
    while( L != NULL ) {
        printf("%d\n", L->item);
        L = L->prox;
    }
}

int main(void) {
    Lista L = NULL;
    exibe(L);
    printf("Maior item da lista: %d\n", maximo(L));

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Erro fatal: a lista esta vazia.

--------------------------------
Process exited after 1.747 seconds with return value 32212254477
Pressione qualquer tecla para continuar. . .
```

Exercício 10

```c
    n->prox = p;
    return n;
}

int pertence(Item x, Lista L) {
    while (L != NULL) {
        if (L->item == x) {
            return 1;
        }
        L = L->prox;
    }
    return 0;
}

void exibe(Lista L) {
    while( L != NULL ) {
        printf("%d\n", L->item);
        L = L->prox;
    }
}

int main(void) {
    printf("Lista:\n");
    Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
    exibe(L);

    int x = 66;
    if (x, pertence(x,L)){
     printf("\nO item %d pertence a lista(L) acima.",x);
    }
    else{
     printf("\nO item %d nao pertence a lista(L) acima.",x);
    }

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Lista:
3
1
5
66
6

O item 66 pertence a lista(L) acima.
--------------------------------
Process exited after 1.135 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 11

```c
    return n;
}

Lista inversa(Lista L) {
    Lista invertida = NULL;
    while (L != NULL) {
        invertida = no(L->item, invertida);
        L = L->prox;
    }
    return invertida;
}


void exibe(Lista L) {
    printf("[");
    while( L != NULL ) {
        printf("%d", L->item);
        if (L->prox != NULL) {
            printf(",");
        }
        L = L->prox;
    }
    printf("]");
}

int main(void) {
    printf("Lista:\n");
    Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
    exibe(L);

    Lista L_invertida = inversa(L);
    printf("\n\nLista in
    exibe(L_invertida);

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Lista:
[3,1,5,66,6]

Lista invertida:
[6,66,5,1,3]
--------------------------------
Process exited after 1.043 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 12

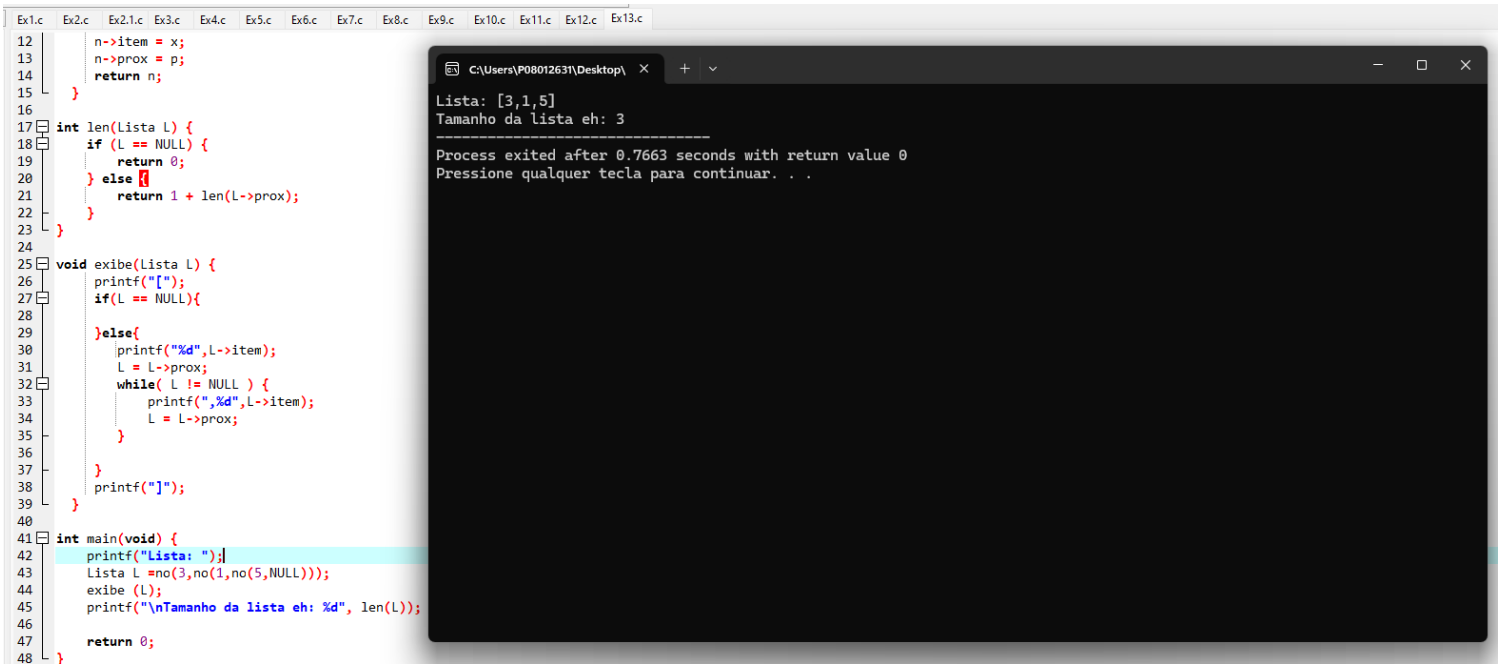
```c
}

Lista inversa(Lista L) {
    Lista invertida = NULL;
    while (L != NULL) {
        invertida = no(L->item, invertida);
        L = L->prox;
    }
    return invertida;
}

void exibe(Lista L) {
    printf("[");
    while( L != NULL ) {
        printf("%d", L->item);
        if (L->prox != NULL) {
            printf(",");
        }
        L = L->prox;
    }
    printf("]");
}

int main(void) {
    int p, u;
    printf("Digite o intervalo de dois numeros (separad
    scanf("%d %d", &p, &u);

    Lista L = intervalo(p, u);
    Lista L_invertida = inversa(L);
    printf("\nIntervalo escolhido: %d...%d\n\n",p,u);
    printf("Lista: \n");
    exibe(L_invertida);

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Digite o intervalo de dois numeros (separados por espaco): -2 3

Intervalo escolhido: -2...3

Lista:
[-2,-1,0,1,2,3]
--------------------------------
Process exited after 3.211 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

```c
    Lista lista = NULL;

    if (p > u) {
        return lista;
    }

    for (int i = p; i <= u; i++) {
        lista = no(i, lista);
    }

    return lista;
}

Lista inversa(Lista L) {
    Lista invertida = NULL;
    while (L != NULL) {
        invertida = no(L->item, invertida);
        L = L->prox;
    }
    return invertida;
}

void exibe(Lista L) {

int main(void) {
    int p, u;
    printf("Digite o intervalo de dois numeros (separados por
    scanf("%d %d", &p, &u);

    Lista L = intervalo(p, u);
    Lista L_invertida = inversa(L);
    printf("\nIntervalo escolhido: %d...%d\n\n",p,u);
    printf("Lista: \n");
    exibe(L_invertida);

    return 0;
```

```
C:\Users\P08012631\Desktop\
Digite o intervalo de dois numeros (separados por espaco): 9 1

Intervalo escolhido: 9...1

Lista:
[]
--------------------------------
Process exited after 4.4 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

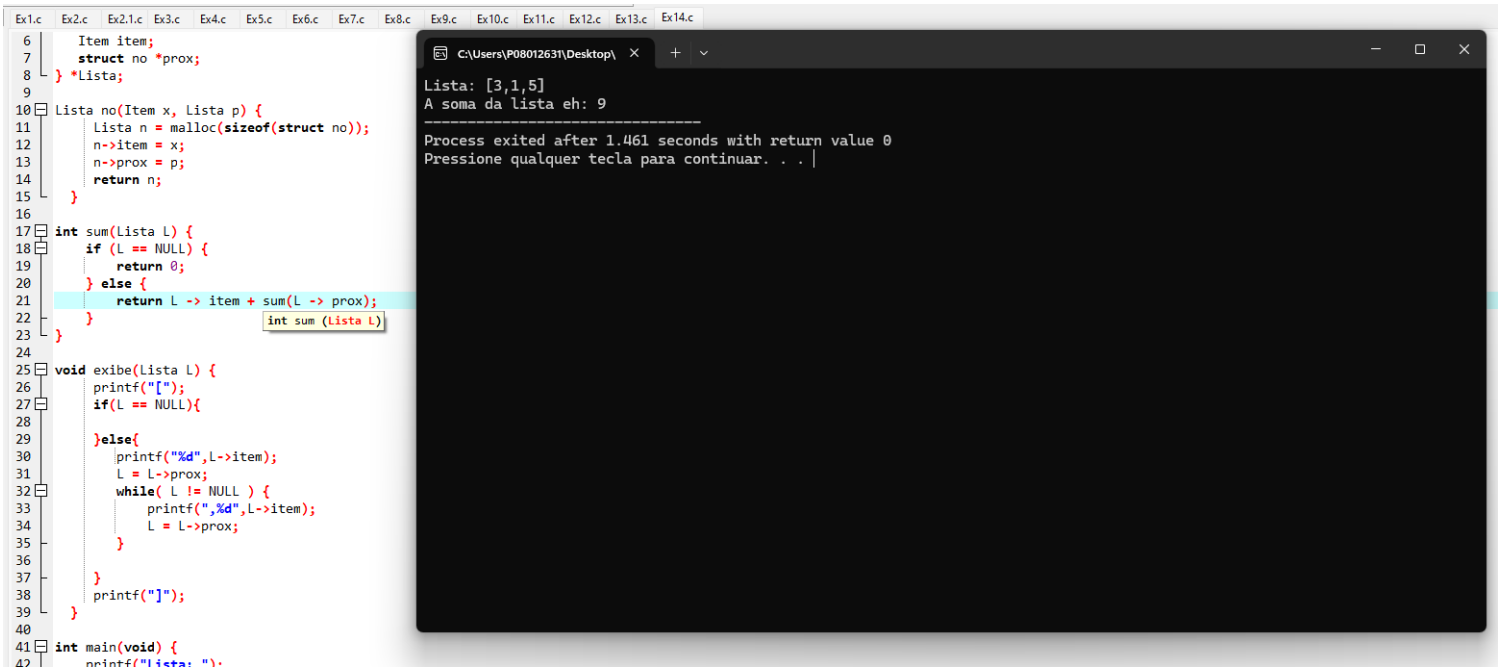
Exercício 13



```
Ex1.c Ex2.c Ex2.1.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex12.c Ex13.c
12      n->item = x;
13      n->prox = p;
14      return n;
15    }
16
17  int len(Lista L) {
18      if (L == NULL) {
19          return 0;
20      } else {
21          return 1 + len(L->prox);
22      }
23  }
24
25  void exibe(Lista L) {
26      printf("[");
27      if(L == NULL){
28
29      }else{
30         printf("%d",L->item);
31         L = L->prox;
32         while( L != NULL ) {
33             printf(",%d",L->item);
34             L = L->prox;
35         }
36
37      }
38      printf("]");
39    }
40
41  int main(void) {
42      printf("Lista: ");
43      Lista L =no(3,no(1,no(5,NULL)));
44      exibe (L);
45      printf("\nTamanho da lista eh: %d", len(L));
46
47      return 0;
48  }
```

```
C:\Users\P08012631\Desktop\
Lista: [3,1,5]
Tamanho da lista eh: 3
--------------------------------
Process exited after 0.7663 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 14



```
Ex1.c Ex2.c Ex2.1.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex12.c Ex13.c Ex14.c
6       Item item;
7       struct no *prox;
8   } *Lista;
9
10  Lista no(Item x, Lista p) {
11      Lista n = malloc(sizeof(struct no));
12      n->item = x;
13      n->prox = p;
14      return n;
15    }
16
17  int sum(Lista L) {
18      if (L == NULL) {
19          return 0;
20      } else {
21          return L -> item + sum(L -> prox);
22      }                          int sum (Lista L)
23  }
24
25  void exibe(Lista L) {
26      printf("[");
27      if(L == NULL){
28
29      }else{
30         printf("%d",L->item);
31         L = L->prox;
32         while( L != NULL ) {
33             printf(",%d",L->item);
34             L = L->prox;
35         }
36
37      }
38      printf("]");
39    }
40
41  int main(void) {
42      printf("Lista: ");
```

```
C:\Users\P08012631\Desktop\
Lista: [3,1,5]
A soma da lista eh: 9
--------------------------------
Process exited after 1.461 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 15

```c
}

int clone(Lista L) {
    if (L == NULL) {
        return 0;
    } else {
        return no(L->item, clone(L->prox));
    }
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }

    }
    printf("]");
}

int main(void) {
    Lista L =no(3,no(1,no(5,NULL)));
    Lista clonada = clone(L);

    printf("Lista original: ");
    exibe(L);
    printf("\n\nCopia da lista: ");
    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Lista original: [3,1,5]

Copia da lista: [3,1,5]
--------------------------------
Process exited after 0.9045 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 16

```c
    for (int i = 0; i < n; i++) {
        Item valor = rand() % m;
        lista = no(valor, lista);
    }

    return lista;
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }

    }
    printf("]");
}

int main(void) {
    int m = 100;
    int n = 0;

    printf("Informe a quantidade de numeros que deseja: ");
    scanf("%d", &n);

    Lista lista_aleatoria = rnd(n, m);
    printf("\nLista de numeros aleatorios com %d itens entre
```

```
C:\Users\P08012631\Desktop\
Informe a quantidade de numeros que deseja: 6

Lista de numeros aleatorios com 6 itens entre 0 e 99:

[11,82,3,59,88,66]
--------------------------------
Process exited after 1.909 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 17

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

Item last(Lista L) {
    if (L == NULL) {
        printf("Erro: a lista esta vazia.\n");
    }
    if (L->prox == NULL) {
        return L->item;
    }
    return last(L->prox);
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }
```

```
C:\Users\P08012631\Desktop\
Lista: [3,1,5,66,6]

O ultimo item da Lista eh: 6
--------------------------------
Process exited after 0.973 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 18

```c
        return 1;
    }
    return in(x, L->prox);
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }
    }
    printf("]");
}

int main(void) {
    int x = 0;
    printf("Informe um numero para verificar se ele pertenc
    scanf("%d", &x);

    printf("\nLista: ");
    Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
    exibe(L);

    if (in(x, L)){
        printf("\n\nO numero %d pertence a Lista.", x);
    }else{
        printf("\n\nO numero %d nao pertence a Lista.", x);
    }

    return 0;
}
```

```
C:\Users\P08012631\Desktop\
Informe um numero para verificar se ele pertence a lista: 5

Lista: [3,1,5,66,6]

O numero 5 pertence a Lista.
--------------------------------
Process exited after 3.647 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 19

```c
Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

Item nth(int n, Lista L) {
    if (n < 1 || L == NULL) {
        printf("Índice invalido ou lista vazia.\n");
    }
    if (n == 1) {
        return L->item;
    }
    return nth(n - 1, L->prox);
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }
    }
    printf("]");
}

int main(void) {
    printf("\nLista: ");
    Lista L = no(3, no(1, no(5, no(66, no(6, NULL)))));
```

```
C:\Users\P08012631\Desktop\
Lista: [3,1,5,66,6]
Informe qual a posicao do numero que deseja visualizar: 4

O 4o. item da lista eh: 66

--------------------------------
Process exited after 9.694 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 20

```c
}

Item minimum(Lista L) {
    if (L == NULL) {
        printf("Lista vazia.\n");
    }
    if (L->prox == NULL) {
        return L->item;
    }
    Item menor = minimum(L->prox);
    return (L->item < menor) ? L->item : menor;
}

void exibe(Lista L) {
    printf("[");
    if(L == NULL){

    }else{
       printf("%d",L->item);
       L = L->prox;
       while( L != NULL ) {
           printf(",%d",L->item);
           L = L->prox;
       }
    }
    printf("]");
}

int main(void) {
    printf("\nLista: ");
    Lista L = no(3, no(1, no(5, no(66, no(6, no(0, no(12, NULL
    exibe(L);

    printf("\n\nO menor item da lista eh: %d\n", minimum(L));

    return 0;
```

```
C:\Users\P08012631\Desktop\
Lista: [3,1,5,66,6,0,12]

O menor item da lista eh: 0

--------------------------------
Process exited after 2.018 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 21

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
   Item item;
   struct no *prox;
} *Lista;

Lista no(Item x, Lista p) {
     Lista n = malloc(sizeof(struct no));
     n->item = x;
     n->prox = p;
     return n;
  }

int sorted(Lista L) {
    if (L == NULL || L->prox == NULL) {
        return 1;
    }
    if (L->item > L->prox->item) {
        return 0;
    }
    return sorted(L->prox);
}

void exibe(Lista L) {
     printf("[");
     if(L == NULL){

     }else{
        printf("%d",L->item);
        L = L->prox;
        while( L != NULL ) {
            printf(",%d",L->item);
            L = L->prox;
```

```
C:\Users\P08012631\Desktop\

Lista: [3,1,5,66,6,0,12]

A lista nao esta ordenada!
--------------------------------
Process exited after 0.8975 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

```c
    if (L == NULL || L->prox == NULL) {
        return 1;
    }
    if (L->item > L->prox->item) {
        return 0;
    }
    return sorted(L->prox);
}

void exibe(Lista L) {
     printf("[");
     if(L == NULL){

     }else{
        printf("%d",L->item);
        L = L->prox;
        while( L != NULL ) {
            printf(",%d",L->item);
            L = L->prox;
        }
     }
     printf("]");
  }

int main(void) {
    printf("\nLista: ");
    Lista L = no(1, no(2, no(3, no(4, no(5, NULL))));
    exibe(L);

    if (sorted(L)){
        printf("\n\nA lista esta ordenada!");
    }else{
        printf("\n\nA lista nao esta ordenada!");
    }

    return 0;
}
```

```
C:\Users\P08012631\Desktop\

Lista: [1,2,3,4,5]

A lista esta ordenada!
--------------------------------
Process exited after 0.8622 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 22

```c
typedef int Item;
typedef struct no {

Lista no(Item x, Lista p) {

int equal(Lista A, Lista B) {
    if (A == NULL && B == NULL) {
        return 1;
    }
    if (A == NULL || B == NULL) {
        return 0;
    }
    if (A->item != B->item) {
        return 0;
    }
    return equal(A->prox, B->prox);
}

void exibe(Lista L) {

int main(void) {
    printf("\nLista A: ");
    Lista A = no(1, no(2, no(3, no(4, no(5, NULL)))));
    exibe(A);

    printf("\nLista B: ");
    Lista B = no(1, no(2, no(3, no(4, no(5, NULL)))));
    exibe(B);

    if (equal(A,B)){
        printf("\n\nAs listas sao iguais!");
    }else{
        printf("\n\nAs listas nao sao iguais!");
    }

    return 0;
}
```

```
C:\Users\P08012631\Desktop\

Lista A: [1,2,3,4,5]
Lista B: [1,2,3,4,5]

As listas sao iguais!
--------------------------------
Process exited after 0.7122 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {

Lista no(Item x, Lista p) {

int equal(Lista A, Lista B) {
    if (A == NULL && B == NULL) {
        return 1;
    }
    if (A == NULL || B == NULL) {
        return 0;
    }
    if (A->item != B->item) {
        return 0;
    }
    return equal(A->prox, B->prox);
}

void exibe(Lista L) {

int main(void) {
    printf("\nLista A: ");
    Lista A = no(3, no(1, no(5, no(66, no(6, no(0, no(12, N
    exibe(A);

    printf("\nLista B: ");
    Lista B = no(1, no(2, no(3, no(4, no(5, NULL)))));
    exibe(B);

    if (equal(A,B)){
        printf("\n\nAs listas sao iguais!");
    }else{
        printf("\n\nAs listas nao sao iguais!");
    }
```

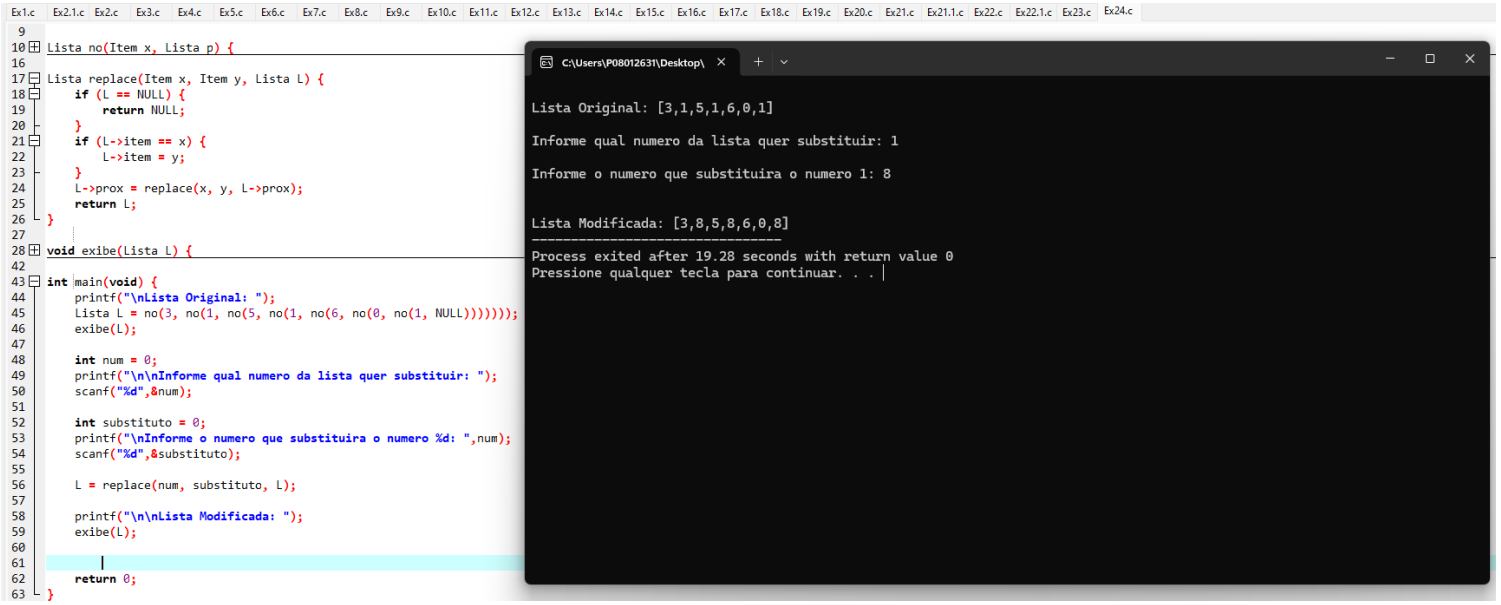
```
C:\Users\P08012631\Desktop\

Lista A: [3,1,5,66,6,0,12]
Lista B: [1,2,3,4,5]

As listas nao sao iguais!
--------------------------------
Process exited after 1.857 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 23

```c
#include <stdlib.h>

typedef int Item;
typedef struct no {

Lista no(Item x, Lista p) {

int count(Item x, Lista L) {
    if (L == NULL) {
        return 0;
    }
    if (L->item == x) {
        return 1 + count(x, L->prox);
    } else {
        return count(x, L->prox);
    }
}

void exibe(Lista L) {

int main(void) {
    printf("\nLista A: ");
    Lista L = no(3, no(1, no(5, no(1, no(6, no(0, no(1, NULL))
    exibe(L);

    int num = 0;
    printf("\n\nInforme um item(numero) para verificar quantas
    scanf("%d",&num);

    int ocorrencias = count(num, L);
    if(count(num,L)){
        printf("\nO item %d ocorre %d vezes na lista.\n", num,
    }else{
        printf("\nO item %d nao ocorre na lista.\n",num);
    }
```

```
C:\Users\P08012631\Desktop\

Lista A: [3,1,5,1,6,0,1]

Informe um item(numero) para verificar quantas vezes ele aparece na lista acima: 1

O item 1 ocorre 3 vezes na lista.

--------------------------------
Process exited after 6.307 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 24



```c
Lista no(Item x, Lista p) {

Lista replace(Item x, Item y, Lista L) {
    if (L == NULL) {
        return NULL;
    }
    if (L->item == x) {
        L->item = y;
    }
    L->prox = replace(x, y, L->prox);
    return L;
}

void exibe(Lista L) {

int main(void) {
    printf("\nLista Original: ");
    Lista L = no(3, no(1, no(5, no(1, no(6, no(0, no(1, NULL)))))));
    exibe(L);

    int num = 0;
    printf("\n\nInforme qual numero da lista quer substituir: ");
    scanf("%d",&num);

    int substituto = 0;
    printf("\nInforme o numero que substituira o numero %d: ",num);
    scanf("%d",&substituto);

    L = replace(num, substituto, L);

    printf("\n\nLista Modificada: ");
    exibe(L);

    return 0;
}
```

```
C:\Users\P08012631\Desktop\

Lista Original: [3,1,5,1,6,0,1]

Informe qual numero da lista quer substituir: 1

Informe o numero que substituira o numero 1: 8


Lista Modificada: [3,8,5,8,6,0,8]
--------------------------------
Process exited after 19.28 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```