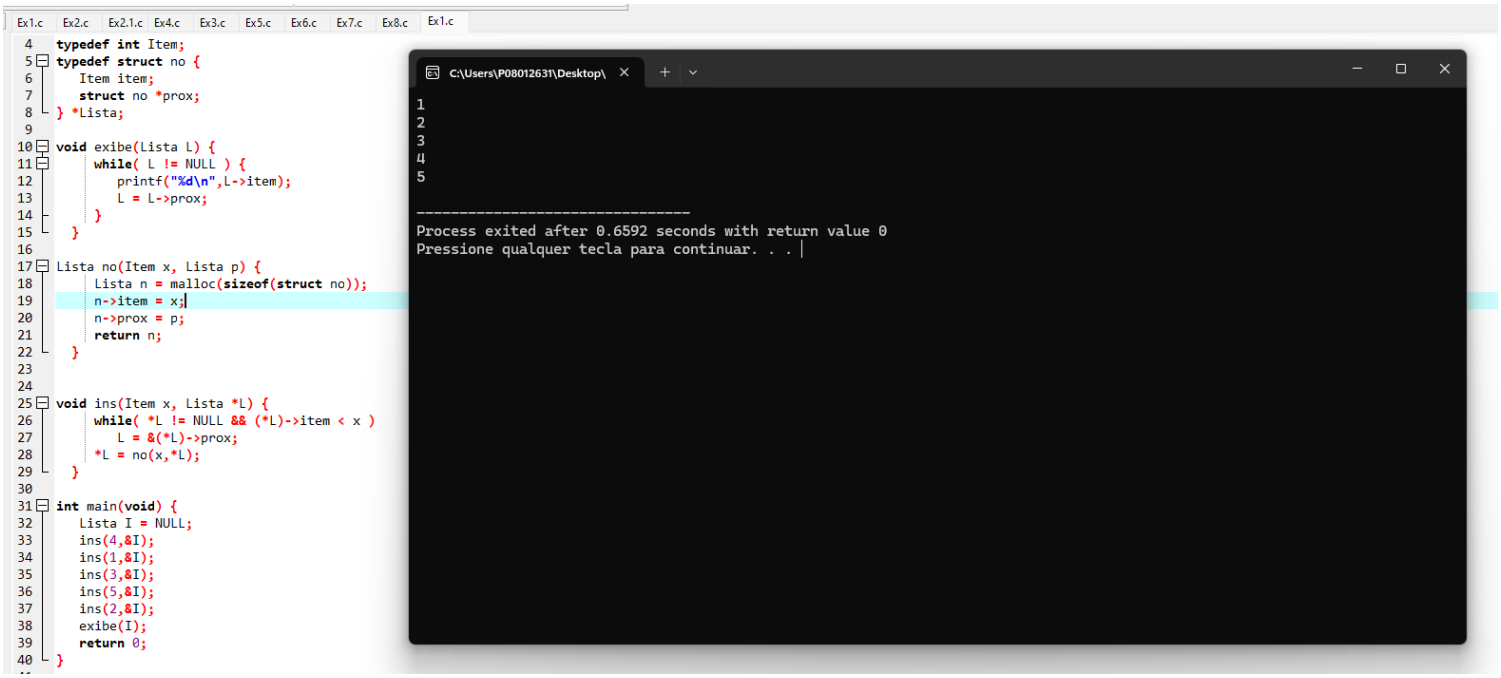
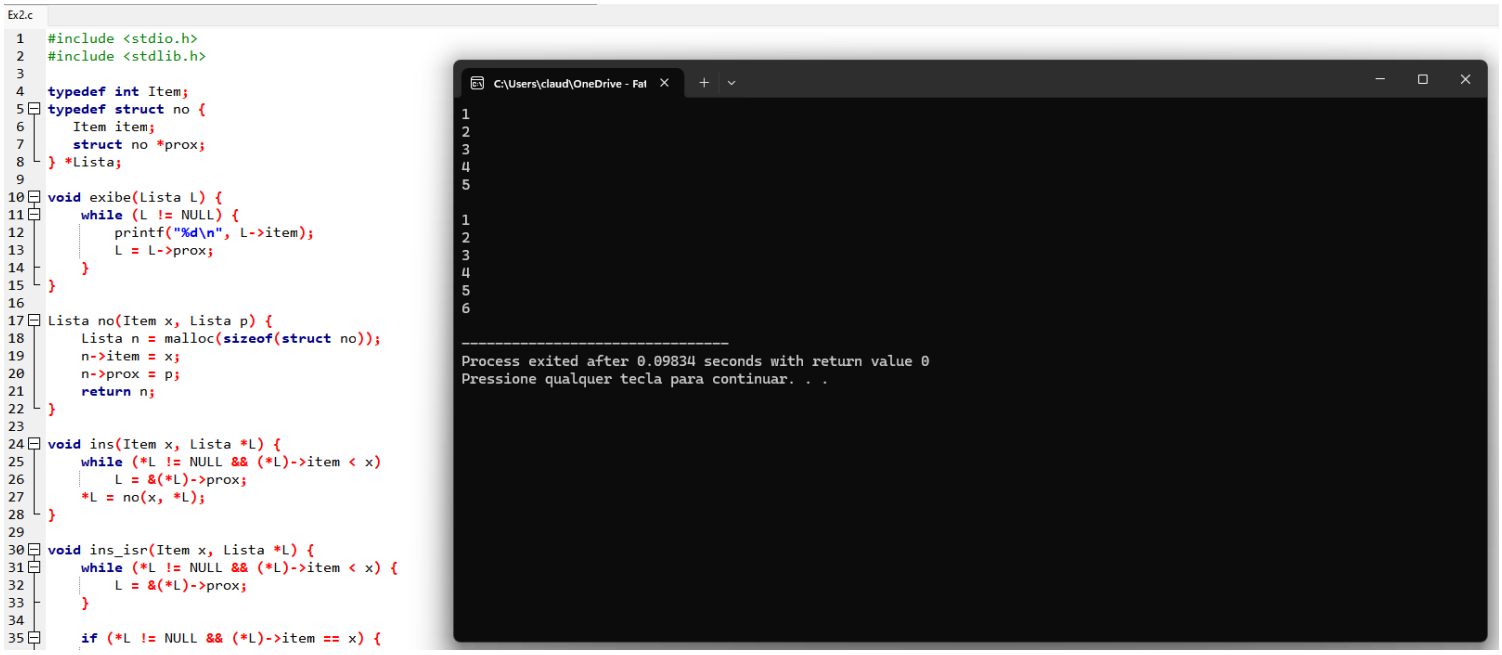
Exercício 1



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

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

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

void ins(Item x, Lista *L) {
    while( *L != NULL && (*L)->item < x )
       L = &(*L)->prox;
    *L = no(x,*L);
}

int main(void) {
   Lista I = NULL;
   ins(4,&I);
   ins(1,&I);
   ins(3,&I);
   ins(5,&I);
   ins(2,&I);
   exibe(I);
   return 0;
}
```

```
C:\Users\P08012631\Desktop\
1
2
3
4
5

--------------------------------
Process exited after 0.6592 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 2



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

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

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

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

void ins(Item x, Lista *L) {
    while (*L != NULL && (*L)->item < x)
        L = &(*L)->prox;
    *L = no(x, *L);
}

void ins_isr(Item x, Lista *L) {
    while (*L != NULL && (*L)->item < x) {
        L = &(*L)->prox;
    }

    if (*L != NULL && (*L)->item == x) {
```

```
C:\Users\claud\OneDrive - Fat
1
2
3
4
5

1
2
3
4
5
6

--------------------------------
Process exited after 0.09834 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 3

Ex2.c Ex4.c Ex3.c

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

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

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

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

void ins_rec(Item x, Lista *L) {
    if (*L == NULL || (*L)->item > x) {
        *L = no(x, *L);
    } else {
        ins_rec(x, &(*L)->prox);
    }
}

int main(void) {
    int num=0;
    Lista I = NULL;
    ins_rec(4, &I);
    ins_rec(1, &I);
```

```
C:\Users\claud\OneDrive - Fat
1
2
3
4
5
10


Informe um numero para inserir na lista:
8

Lista ordenada:
1
2
3
4
5
8
10

--------------------------------
Process exited after 5.492 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 4

Ex2.c Ex3.c

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

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

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

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

void ins(Item x, Lista *L) {
    while (*L != NULL && (*L)->item < x)
        L = &(*L)->prox;
    *L = no(x, *L);
}

void ins_rsr(Item x, Lista *L) {
    if (*L == NULL || (*L)->item > x) {
        *L = no(x, *L);
    } else if ((*L)->item < x) {
        ins_rsr(x, &(*L)->prox);
    }
```

```
C:\Users\claud\OneDrive - Fat
1
2
3
4
5

1
2
3
4
5
6

--------------------------------
Process exited after 0.08435 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 5

```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 ins_rec(Item x, Lista *L) {
    if (*L == NULL || (*L)->item > x) {
        *L = no(x, *L);
    } else {
        ins_rec(x, &(*L)->prox);
    }
}

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

int main(void) {
    Lista I = NULL;
    ins_rec(4, &I);
    ins_rec(1, &I);
    ins_rec(3, &I);
```

```
C:\Users\claud\OneDrive - Fat
1
2
3
4
5

--------------------------------
Process exited after 0.07709 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 6

```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 ins_rec(Item x, Lista *L) {
    if (*L == NULL || (*L)->item < x) {
        *L = no(x, *L);
    } else {
        ins_rec(x, &(*L)->prox);
    }
}

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

int main(void) {
    Lista I = NULL;
    ins_rec(4, &I);
    ins_rec(1, &I);
    ins_rec(3, &I);
```

```
C:\Users\claud\OneDrive - Fat
5
4
3
2
1

--------------------------------
Process exited after 0.1027 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 7

Ex1.c Ex3.c Ex2.c Ex4.c Ex5.c Ex6.c Ex7.c

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

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

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

void ins(Item x, Lista *L) {
     while( *L != NULL && (*L)->item < x )
        L = &(*L)->prox;
     *L = no(x,*L);
  }

void rem(Item x, Lista *L) {
     while( *L != NULL && (*L)->item < x )
        L = &(*L)->prox;
     if( *L == NULL || (*L)->item > x ) return;
     Lista n = *L;
     *L = n->prox;
     free(n);
  }

int main(void) {
   Lista I = NULL;
   ins(4,&I); ins(1,&I); ins(3,&I); ins(5,&I); ins(2,&I);
```

```
C:\Users\claud\OneDrive - Fat
1
2
4
5

--------------------------------
Process exited after 0.1165 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 8

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

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

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

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

void ins(Item x, Lista *L) {
     while( *L != NULL && (*L)->item < x )
        L = &(*L)->prox;
     *L = no(x,*L);
  }

void rem(Item x, Lista *L) {
     while( *L != NULL && (*L)->item < x )
        L = &(*L)->prox;
     if( *L == NULL || (*L)->item > x ) retu
     Lista n = *L;
     *L = n->prox;
```

```
C:\Users\claud\OneDrive - Fat
Lista antes de remover:
1
2
3
3
3
4
5

Lista depois de remover todas as ocorrencias de 3:
1
2
4
5

--------------------------------
Process exited after 0.1074 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 9

```c
    free(n);
  }

void rem_todo(Item x, Lista *L) {
    while (*L != NULL) {
        if ((*L)->item < x) {
            L = &(*L)->prox;
        } else if ((*L)->item == x) {
            Lista n = *L;
            *L = n->prox;
            free(n);
        } else {
            break;
        }
    }
}

int pert(Item x, Lista L) {
    while( L != NULL && L->item < x )
        L = L->prox;
    return (L != NULL && L->item == x);
  }


int main(void) {
   Lista I = NULL;
   ins(4,&I);
   ins(1,&I);
   ins(3,&I);
   ins(2,&I);
   printf("%d\n", pert(5,I));
   printf("%d\n", pert(3,I));
   return 0;
}

```

```
C:\Users\claud\OneDrive - Fat
0
1

--------------------------------
Process exited after 0.1045 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 10

```c
            *L = n->prox;
            free(n);
        } else {
            break;
        }
    }
}

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

int main(void) {
    Lista I = NULL;
    ins(4, &I);
    ins(1, &I);
    ins(3, &I);
    ins(2, &I);


    printf("Usando pertinencia recurs
    printf("5 esta na lista? %d\n", p
    printf("3 esta na lista? %d\n", p

    return 0;
}

```

```
C:\Users\claud\OneDrive - Fat
Usando pertinencia recursiva (0=Falso, 1=Verdadeiro):
5 esta na lista? 0
3 esta na lista? 1

--------------------------------
Process exited after 0.1258 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 11

```c
    *L = no(x, *L);
}

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

int main(void) {
    Lista I = NULL;
    ins(1, &I);
    ins(2, &I);
    ins(3, &I);
    ins(4, &I);

    printf("Lista antes de verificar se está
    exibe(I);

    printf("A lista está ordenada de forma cr

    ins(3, &I);

    printf("\nLista depois de inserir um valo
    exibe(I);

    printf("A lista está ordenada de forma cr

    return 0;
}
```

```
C:\Users\claud\OneDrive - Fat
Lista antes de verificar se estß ordenada:
1
2
3
4
A lista estß ordenada de forma crescente? 1

Lista depois de inserir um valor fora de ordem:
1
2
3
3
4
A lista estß ordenada de forma crescente? 1

--------------------------------
Process exited after 0.1266 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 12

```c
}

void ins(Item x, Lista *L) {
    while (*L != NULL && (*L)->item < x)
        L = &(*L)->prox;
    *L = no(x, *L);
}

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

int main(void) {
    Lista I = NULL;
    ins(1, &I);
    ins(2, &I);
    ins(3, &I);
    ins(4, &I);

    printf("Lista antes de verificar se está ordenada:\n")
    exibe(I);

    printf("A lista está ordenada de forma crescente? %d\r

    ins(3, &I);

    printf("\nLista depois de inserir um valor fora de ord
    exibe(I);
```

```
C:\Users\claud\OneDrive - Fat
Lista antes de verificar se estß ordenada:
1
2
3
4
A lista estß ordenada de forma crescente? 1

Lista depois de inserir um valor fora de ordem:
1
2
3
3
4
A lista estß ordenada de forma crescente? 0

--------------------------------
Process exited after 0.1083 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 13

```
Ex1.c Ex3.c Ex2.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex12.c Ex13.c
```

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

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

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

void ins(Item x, Lista *L) {
    while (*L != NULL && (*L)->item > x)
        L = &(*L)->prox;
    *L = no(x, *L);
}

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

int main(void) {
    Lista I = NULL;
```

```
C:\Users\claud\OneDrive - Fal
Lista antes de verificar se esta ordenada:
4
3
2
1
A lista esta ordenada de forma decrescente? 1

Lista depois de inserir um valor fora de ordem:
4
3
3
2
1
A lista esta ordenada de forma decrescente? 1

--------------------------------
Process exited after 0.1265 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 14

```
Ex1.c Ex3.c Ex2.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex12.c Ex13.c Ex14.c
```

```c
    *L = no(x, *L);
}

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

int main(void) {
    Lista I = NULL;
    ins(4, &I);
    ins(3, &I);
    ins(2, &I);
    ins(1, &I);

    printf("Lista antes de verificar se esta orde
    exibe(I);

    printf("A lista esta ordenada de forma decres

    ins(3, &I);

    printf("\nLista depois de inserir um valor fo
    exibe(I);

    printf("A lista esta ordenada de forma decres

    return 0;
}
```

```
C:\Users\claud\OneDrive - Fal
Lista antes de verificar se esta ordenada:
4
3
2
1
A lista esta ordenada de forma decrescente? 1

Lista depois de inserir um valor fora de ordem:
4
3
3
2
1
A lista esta ordenada de forma decrescente? 0

--------------------------------
Process exited after 0.1685 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```