```c
// Implementa uma lista de números com um array de tamanho fixo

#include <stdio.h>

int main(void)
{
    // Lista de tamanho 3
    int lista[3];

    // Inicializa lista com números
    lista[0] = 1;
    lista[1] = 2;
    lista[2] = 3;

    // Imprime lista
    for (int i = 0; i < 3; i++)
    {
        printf("%i\n", lista[i]);
    }
}
```

```c
// Implementa uma lista de números com um array de tamanho dinâmico

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

int main(void)
{
    // Lista de tamanho 3
    int *lista = malloc(3 * sizeof(int));
    if (lista == NULL)
    {
        return 1;
    }

    // Inicializa lista de tamanho 3 com números
    lista[0] = 1;
    lista[1] = 2;
    lista[2] = 3;

    // Lista de tamanho 4
    int *temporario = malloc(4 * sizeof(int));
    if (temporario == NULL)
    {
        return 1;
    }

    // Copia lista de tamanho 3 para lista de tamanho 4
    for (int i = 0; i < 3; i++)
    {
        temporario[i] = lista[i];
    }

    // Adiciona número à lista de tamanho 4
    temporario[3] = 4;

    // Libera lista de tamanho 3
    free(lista);

    // Grava lista de tamanho 4 no ponteiro *lista
    lista = temporario;

    // Imprime lista
    for (int i = 0; i < 4; i++)
    {
        printf("%i\n", lista[i]);
```

```
46        }
47
48        // Libera lista
49        free(lista);
50    }
```

```c
// Implementa uma lista de números com um array de tamanho dinâmico usando realloc

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

int main(void)
{
    // Lista de tamanho 3
    int *lista = malloc(3 * sizeof(int));
    if (lista == NULL)
    {
        return 1;
    }

    // Inicializa lista de tamanho 3 com números
    lista[0] = 1;
    lista[1] = 2;
    lista[2] = 3;

    // Redimensiona lista para que tenha tamanho 4
    int *temporario = realloc(lista, 4 * sizeof(int));
    if (temporario == NULL)
    {
        return 1;
    }
    lista = temporario;

    // Adiciona número à lista
    lista[3] = 4;

    // Imprime lista
    for (int i = 0; i < 4; i++)
    {
        printf("%i\n", lista[i]);
    }

    // Libera lista
    free(lista);
}
```

```c
// Implementa uma lista de números usando uma lista encadeada

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

// Representa um nodo
typedef struct nodo
{
    int numero;
    struct nodo *proximo;
}
nodo;

int main(void)
{
    // Lista de tamanho 0
    nodo *lista = NULL;

    // Adiciona número à lista
    nodo *n = malloc(sizeof(nodo));
    if (n == NULL)
    {
        return 1;
    }
    n->numero = 1;
    n->proximo = NULL;
    lista = n;

    // Adiciona número à lista
    n = malloc(sizeof(nodo));
    if (n == NULL)
    {
        return 1;
    }
    n->numero = 2;
    n->proximo = NULL;
    lista->proximo = n;

    // Adiciona número à lista
    n = malloc(sizeof(nodo));
    if (n == NULL)
    {
        return 1;
    }
    n->numero = 3;
```

```c
    n->proximo = NULL;
    lista->proximo->proximo = n;

    // Imprime lista
    for (nodo *temporario = lista; temporario != NULL; temporario = temporario->proximo)
    {
        printf("%i\n", temporario->numero);
    }

    // Libera lista
    while (lista != NULL)
    {
        nodo *temporario = lista->proximo;
        free(lista);
        lista = temporario;
    }
}
```