unisepe EDUCACIONAL

UNIDADE II

Programação Estruturada

unisepe EDUCACIONAL

UNIDADE II

CAPÍTULO 5 – LISTAS LINEARES

No término deste capítulo, você deverá saber:

- ✓ Conceitos básicos e definições relacionadas a listas lineares
- ✓ Implementar listas lineares utilizando alocação dinâmica e alocação estática
- ✓ O funcionamento das operações de criação, inserção, remoção e busca em listas lineares
- ✓ Tipos de listas: encadeada, duplamente encadeada e circular

Considere que o histórico de vendas de sacos de ração esteja apresentado na Tabela 1.

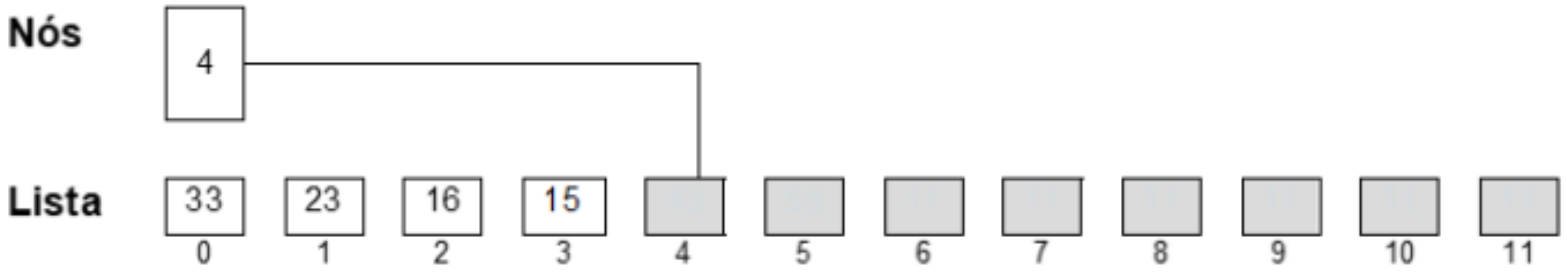
Tabela 1 – Histórico de venda de saco de ração de 2019

Código do mês	Mês	Quantidade
0	Janeiro	33
1	Fevereiro	23
2	Março	16
3	Abril	15
4	Maio	43
5	Junho	58
6	Julho	11
7	Agosto	64
8	Setembro	76
9	Outubro	27
10	Novembro	79
11	Dezembro	78

5.2 Implementação de lista estática

Tanto o processo de inserção quanto o processo de remoção têm um custo computacional bastante elevado nesse tipo de lista estática. Esse tipo de lista é aconselhado para situações bem específicas, tais como:

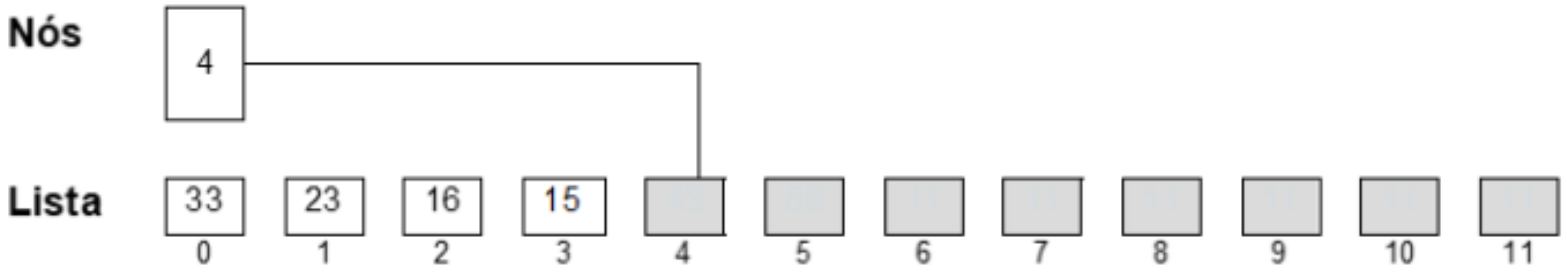
- Listas pequenas, pois esse tipo de lista é fácil de implementar;
- Listas nas quais a inserção e remoção forem realizadas somente no final, assim não há deslocamentos de elementos;
- Listas nas quais a busca for a operação mais frequente, pois a busca é rápida e o tempo de acesso a qualquer um dos elementos é sempre o mesmo.



5.2 Implementação de lista estática

Tanto o processo de inserção quanto o processo de remoção têm um custo computacional bastante elevado nesse tipo de lista estática. Esse tipo de lista é aconselhado para situações bem específicas, tais como:

- Listas pequenas, pois esse tipo de lista é fácil de implementar;
- Listas nas quais a inserção e remoção forem realizadas somente no final, assim não há deslocamentos de elementos;
- Listas nas quais a busca for a operação mais frequente, pois a busca é rápida e o tempo de acesso a qualquer um dos elementos é sempre o mesmo.



Quadro 2 – Código da função de inserção em lista estática

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #define NUMNODES 12
4
5  struct no{
6  int quantidadeRacao;
7  char nomeMes[30];
8  };
9  struct no nos[NUMNODES];
10
11  int main(){
12    int qtdeNos=0;
13    char op;
14
15    do{
16    printf("Você deseja inserir um novo elemento?");
17    scanf("%c",&op);
18
19    if(op == 'S'){
20          printf("Informe o nome do mês:");
21          scanf("%s",&nos[qtdeNos].nomeMes);
22          printf("Informe a quantidade de racao:");
23          scanf("%d",&nos[qtdeNos].quantidadeRacao);
24          fflush(stdin);
25          qtdeNos++;
26      }
27    }while(op != 'N' && qtdeNos != 12);
28
29  return 0;
30    }
```

```c
#include<stdio.h>
#include<stdlib.h>
#define NUMNODES 12

struct no{
int quantidadeRacao;
char nomeMes[30];
};
struct no nos[NUMNODES];

int main(){
   int qtdeNos=0;
   char op;

```

```
14
15    do{
16    printf("Você deseja inserir um novo elemento?");
17    scanf("%c",&op);
18
19    if(op == 'S'){
20          printf("Informe o nome do mês:");
21          scanf("%s",&nos[qtdeNos].nomeMes);
22          printf("Informe a quantidade de racao:");
23          scanf("%d",&nos[qtdeNos].quantidadeRacao);
24          fflush(stdin);
25          qtdeNos++;
26      }
27    }while(op != 'N' && qtdeNos != 12);
28
29  return 0;
30     }
```

Quadro 3 – Código da função de busca de um nó pelo campo “nomeMes”

```
1  void busca_campo_nomeMes(){
2      int i=0;
3      char chave[30]="outubro";
4
5      for(i=0;i<NUMNODES;i++){
6          if(strcmp(chave,nos[i].nomeMes) == 0)
7              printf("Sacos de racao: %d",nos[i].quantidadeRacao);
8      }
9  }
```

Quadro 4 – Código da função de remoção de um nó

```
1  #define REMOVIDO -1
2  void busca_campo_nomeMes(){
3      int i=0;
4      char chave[30]="julho";
5
6      for(i=0;i<NUMNODES;i++){
7          if(strcmp(chave,nos[i].nomeMes) == 0)
8              nos[i].quantidadeRacao = REMOVIDO;
9      }
10 }
```

UNIDADE II

CAPÍTULO 5
LISTAS LINEARES

unisepe EDUCACIONAL

ATIVIDADE

- Utilizando o C++ implemente e teste o código acima.
- Após testa-lo, crie o seguinte menu com as operações:

    1 - MOSTRAR LISTA

    2 - ADICIONAR NA LISTA

    3 - REMOVER DA LISTA

    4 - SAIR DO PROGRAMA

Envie por email: ads@faculdadesaolourenco.com.br

Quadro 5 – Arquivo *lista.h*

```
1  #define NUMNODES 12
2
3    typedef struct NO{
4    int quantidadeRacao;
5    char nomeMes[30];
6    }no;
7
8   no nos[NUMNODES];
9
10  void inserir_no_inicio();
11  void inserir_no_meio();
12  void inserir_no_fim();
13  void busca_no();
14  void remove_no_inicio();
15  void remove_no_meio();
16  void remove_no_fim();
17  void menu();
18  void cria_novo();
19  int verifica_lista();
20  int lista_vazia();
21  int encontra_posicao(char* ant)
22  void desloca_elementos_insercao(int inicio, int fim);
23  int desloca_elementos_remoção(int inicio, int fim);
24  void imprime();
```

Quadro 5 – Arquivo *lista.h*

```
1  #define NUMNODES 12
2
3     typedef struct NO{
4     int quantidadeRacao;
5     char nomeMes[30];
6     }no;
7
8   no nos[NUMNODES];
9
```

```
10  void inserir_no_inicio();
11  void inserir_no_meio();
12  void inserir_no_fim();
13  void busca_no();
14  void remove_no_inicio();
15  void remove_no_meio();
16  void remove_no_fim();
17  void menu();
18  void cria_novo();
19  int verifica_lista();
20  int lista_vazia();
21  int encontra_posicao(char* ant)
22  void desloca_elementos_insercao(int inicio, int fim);
23  int desloca_elementos_remoção(int inicio, int fim);
24  void imprime();
```

UNIDADE II

UNIDADE II

ATIVIDADE

Com base nas assinaturas definidas no arquivo.h, nós já podemos criar o arquivo de implementação lista.c.

Utilizando o C++ implemente e teste o código da páginas da apostila:

PAGINA 56 → Quadro 5 – Arquivo lista.h

PAGINA 58 → Arquivo lista.c

Envie por email: ads@faculdadesaolourenco.com.br