INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 12

1. Implemente um TAD Pilha como lista encadeada. O TAD deve exportar e implementar as seguintes funções:

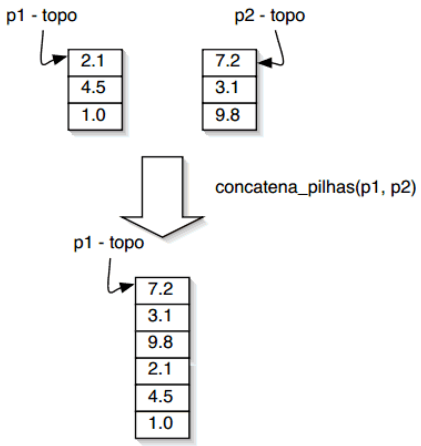
```
Pilha* pilha_cria();
void pilha_push(Pilha* p, float v);
float pilha_pop(Pilha* p);
int pilha_vazia(Pilha* p);
void pilha_libera(Pilha* p);
```

Utilizando o TAD Pilha, implemente as seguintes funções:

- `retiraImpares` – A função recebe uma pilha e retire todos os elementos ímpares dessa pilha, mantendo a ordem original dos elementos pares da pilha. Essa função tem o seguinte protótipo:

```
void retiraImpares(Pilha* p);
```

- `concatenaPilhas` – A função recebe duas pilhas, p1 e p2, e passa todos os elementos da pilha p2 para o topo da pilha p1. A figura a seguir ilustra a concatenação de pilhas:



Ao final da concatenação, a pilha p2 deve estar vazia e a pilha p1 deve conter todos os elementos das duas pilhas. A função deve obedecer o seguinte protótipo:

```
void concatenaPilhas(Pilha* p1, Pilha* p2);
```

Após implementar as funções, crie a função principal do seu programa. Nela, você deve criar duas Pilhas com pelo menos 6 números. Em seguida o programa deve remover todos os números impares da primeira Pilha e concatenar a Pilha resultante com a segunda Pilha. Por ultimo, a Pilha concatenada deve ser exibida na tela.

Após testar o seu programa, crie uma nova implementação do TAD Pilha usando vetores. Em seguida, teste novamente as funções criadas usando a nova implementação do TAD.

2. Continue a implementação do programa criado no exercício anterior, adicionando a ele um TAD Lista que deve exportar e implementar as seguintes funções:

```
Elemento* lista_insere(Elemento * n, float a);
Elemento* lista_retira(Elemento * n, float a);
void lista_libera(Elemento * n);
void lista_imprime(Elemento * n);
Elemento* lista_acessa_prox(Elemento * n);
float lista_acessa_info(Elemento * n);
void lista_atualiza_info(Elemento * n, float a);
```

Utilizando o TAD Pilha e o TAD Lista, implemente a seguinte função:

- `inverteLista` – A função retorna a inversa de uma Lista de números float, usando uma pilha como estrutura auxiliar. Essa função tem o seguinte protótipo:

```
Elemento* inverteLista(Elemento *n);
```

Após implementar a função, modifique a função principal do seu programa criando uma nova Lista e adicionando alguns valores dentro da lista. Em seguida, utilize a função `inverteLista` para inverter a lista. Por ultimo, exiba a lista invertida na tela.