

Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Frequência F4

Arquivo Pilha.cpp

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

typedef struct
{
    char letra;

} Item;

typedef struct st_celula
{
    Item* item;
    struct st_celula* proximo;

} Celula;

typedef struct
{
    Celula* topo;
    int tamanho;

} Pilha;

int compara(Pilha*, Pilha*);
Pilha* cria();
Item* POP(Pilha*);
int PUSH(Item*, Pilha*);
int exhibe(Pilha*);
Pilha* inverte(Pilha*);

int main(void)
{
```

```

Pilha* p = cria();
Item item;
exibe(p); // vazia
// um loop para carregar uns itens
printf("Insere de 'A' a 'Z'\n");
for (item.letra = 'A'; item.letra <= 'Z'; item.letra += 1) PUSH(&item, p);
exibe(p);
int n = p->tamanho / 2;
printf("Desempilhando %d elementos\n", n);
for (int i = 0; i < n; i += 1)
    printf("%c ", (POP(p))>letra);
printf("\n");

printf("Pilha restante\n");
exibe(p);

printf("Inverte a Pilha\n");
Pilha* inv_p = inverte(p);
exibe(inv_p);
printf("Pilha original\n");
exibe(p);

// cria um apilha igual a inv_p pra comparar
printf("\n\nComparando pilhas:\n\n");
Pilha* outra = cria();
exibe(outra);
exibe(inv_p);
if (compara(inv_p, outra))
    printf("As pilhas sao iguais\n");
else
    printf("As pilhas sao diferentes\n");
printf("\nCarregando a pilha que estava vazia: \n");
Insere de 'B' a 'M'\n");
for (item.letra = 'M'; item.letra >= 'B'; item.letra -= 1) PUSH(&item, outra);
printf("\nCompara de novo:\n");
exibe(outra);
exibe(inv_p);
if (compara(inv_p, outra))
    printf("As pilhas sao iguais\n");
else
    printf("As pilhas sao diferentes\n");
printf("\nInsere a letra que faltava e compara de novo:\n");
item.letra = 'A';
PUSH(&item, outra);
exibe(outra);
exibe(inv_p);
if (compara(inv_p, outra))
    printf("As pilhas sao iguais\n");

```

```

else
    printf("As pilhas sao diferentes\n");

return 0;
}

int compara(Pilha* uma, Pilha* outra)
{ // retorna zero se 'uma' difere de 'outra', ou 1
  if (uma->tamanho != outra->tamanho) return 0; // claro
  Celula* pA = uma->topo;
  Celula* pB = outra->topo;
  // pA e pB apontam para o topo de cada pilha
  // olha um por um e se for diferente ja era
  for (int i = 0; i < uma->tamanho; i += 1)
  {
    if (pA->item->letra != pB->item->letra) return 0;
    pA = pA->proximo; pB = pB->proximo; // avanca
  }
  return 1;
}

```

```

Pilha* cria()
{
  Pilha* nova = (Pilha*)malloc(sizeof(Pilha));
  nova->tamanho = 0;
  nova->topo = NULL;
  return nova;
};

```

```

Item* POP(Pilha* pilha)
{
  if (pilha->tamanho < 1) return NULL; // vazia
  Item* valor = (Item*)malloc(sizeof(Item));
  *valor = *(pilha->topo->item);
  Celula* topo = pilha->topo; // salva para nao perder
  pilha->topo = topo->proximo;
  pilha->tamanho -= 1; // um a menos;
  free(topo->item); // adeus item
  free(topo); // adeus celula
  return valor;
}

```

```

int PUSH(Item* item, Pilha* pilha)
{

```

```

Celula* nova = (Celula*)malloc(sizeof(Celula));
Item* pltem = (Item*)malloc(sizeof(Item));
*(pltem) = *item; // copia tudo
nova->item = pltem; // o ponteiro
nova->proximo = pilha->topo;
pilha->topo = nova;
pilha->tamanho += 1;
return 0;
} // empilha()

```

```

int     exhibe(Pilha* pilha)
{
    printf("Pilha tem %d elementos\n", pilha->tamanho);
    Celula* p = pilha->topo;
    while (p != NULL)
    {
        printf("%c' ", p->item->letra);
        p = p->proximo;
    };
    printf("]\n");
    return 0;
};

```

```

Pilha* inverte(Pilha* origem)
{
    Pilha* destino = cria();
    // enquanto tem algo na origem insere na outra
    Item* item = POP(origem);
    while (item != NULL)
    {
        PUSH(item, destino);
        item = POP(origem);
    }
    return destino;
};

```

Prints da execução

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4
5 using namespace std;
6
7 struct string {
8     char * letra;
9 };
10
11 struct stack {
12     string * letra;
13     int tamanho;
14 };
15
16 int main() {
17     string s;
18     int n;
19     cout << "Digite o tamanho da string: ";
20     cin >> n;
21     s.letra = new char[n];
22     for (int i = 0; i < n; i++) {
23         s.letra[i] = ' ';
24     }
25     cout << "Digite a string: ";
26     for (int i = 0; i < n; i++) {
27         cin >> s.letra[i];
28     }
29     cout << endl;
30     stack s1;
31     s1.letra = new string[n];
32     s1.tamanho = 0;
33     for (int i = 0; i < n; i++) {
34         s1.letra[i] = s.letra[i];
35     }
36     cout << "Comparando pilhas: ";
37     if (s1.tamanho == 0) {
38         cout << "A pilha esta vazia." << endl;
39     } else {
40         cout << "A pilha tem " << s1.tamanho << " elementos." << endl;
41         for (int i = 0; i < s1.tamanho; i++) {
42             cout << s1.letra[i] << " ";
43             if (i % 10 == 9) {
44                 cout << endl;
45             }
46         }
47         cout << endl;
48     }
49     if (s1.tamanho == 0) {
50         cout << "Carregando a pilha que estava vazia: Insere de '0' a 'N'." << endl;
51     }
52     cout << "Processo executado com sucesso." << endl;
53     return 0;
54 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED1\F4\Project1.exe
- Output Size: 130,968,203,128 KiB
- Compilation Time: 0.19s

Line: 12 Col: 2 Sel: 0 Lines: 167 Length: 3993 Insert Done parsing in 0.125 seconds

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4
5 using namespace std;
6
7 struct string {
8     char * letra;
9 };
10
11 struct stack {
12     string * letra;
13     int tamanho;
14 };
15
16 int main() {
17     string s;
18     int n;
19     cout << "Digite o tamanho da string: ";
20     cin >> n;
21     s.letra = new char[n];
22     for (int i = 0; i < n; i++) {
23         s.letra[i] = ' ';
24     }
25     cout << "Digite a string: ";
26     for (int i = 0; i < n; i++) {
27         cin >> s.letra[i];
28     }
29     cout << endl;
30     stack s1;
31     s1.letra = new string[n];
32     s1.tamanho = 0;
33     for (int i = 0; i < n; i++) {
34         s1.letra[i] = s.letra[i];
35     }
36     cout << "Comparando pilhas: ";
37     if (s1.tamanho == 0) {
38         cout << "A pilha esta vazia." << endl;
39     } else {
40         cout << "A pilha tem " << s1.tamanho << " elementos." << endl;
41         for (int i = 0; i < s1.tamanho; i++) {
42             cout << s1.letra[i] << " ";
43             if (i % 10 == 9) {
44                 cout << endl;
45             }
46         }
47         cout << endl;
48     }
49     if (s1.tamanho == 0) {
50         cout << "Carregando a pilha que estava vazia: Insere de '0' a 'N'." << endl;
51     }
52     cout << "Processo executado com sucesso." << endl;
53     return 0;
54 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED1\F4\Project1.exe
- Output Size: 130,968,203,128 KiB
- Compilation Time: 0.19s

Line: 12 Col: 2 Sel: 0 Lines: 167 Length: 3993 Insert Done parsing in 0.125 seconds