

```

1  #include "pilha.h"
2  #include <stdio.h>
3
4  int main(void) {
5      int num;
6      Pilha A = pilha(10);
7      Pilha B = pilha(10);
8
9      printf("Digite 10 numeros aleatorios:\n");
10     for (int i = 0; i < 10; i++) {
11         scanf("%d", &num);
12         empilha(num, A);
13     }
14     printf("Vamos ordenar...\n");
15
16     while (!vaziap(A)) {
17         int item_de_A = desempilha(A);
18
19         while (!vaziap(B) && topo(B) > item_de_A) {
20             empilha(desempilha(B), A);
21         }
22         empilha(item_de_A, B);
23     }
24     while (!vaziap(B)) {
25         empilha(desempilha(B), A);
26     }
27
28     printf("Pilha ordenada: ");
29     while (!vaziap(A)) {
30         printf("%d ", desempilha(A));
31     }
32     printf("\n");
33
34     destroip(&A);
35     destroip(&B);
36     return 0;
37 }

```

Saída

```

1: morelli@morelli-archlinux:~/Documentos/Fatec/EstruturaDados/Pilha/Tarefa
~/EstruturaDados/Pilha > cd Tarefa/

~/Pilha/Tarefa > ./output/
...1* ...2* ...3* ...5* ...6*

~/Pilha/Tarefa > ./output/exercicio1
Digite 10 numeros aleatorios:
2
5
3
6
2

6
7
5
4
3
Vamos ordenar...
Pilha ordenada: 2 2 3 3 4 5 5 6 6 7

```

Tarefa 2

```

1  #include "pilha.h"
2  #include <stdio.h>
3  int main(void) {
4      int num;
5      Pilha A = pilha(10);
6      Pilha B = pilha(10);
7      printf("Digite 10 numeros aleatorios:\n");
8      for (int i = 0; i < 10; i++) {
9          if (scanf("%d", &num) != 1) {
10             printf("Erro de leitura.\n");
11             break;
12         }
13         empilha(num, A);
14     }
15     printf("Vamos ordenar inversamente e garantir a unicidade...\n");
16     while (!vaziap(A)) {
17         int item_de_A = desempilha(A);
18
19         while (!vaziap(B) && topo(B) < item_de_A) {
20             empilha(desempilha(B), A);
21         }
22         empilha(item_de_A, B);
23     }
24     printf("Pilha ordenada inversamente (decrescente) e sem repetidos: \n");
25     while (!vaziap(B)) {
26         int elemento_final = desempilha(B);
27         empilha(elemento_final, A);

```

```

28     }
29     printf("\n");
30     while (!vaziap(A)) {
31         int eEquals = desempilha(A);
32         if (!vaziap(A) && eEquals == topo(A)) {
33             continue;
34         }
35         printf("%d ", eEquals);
36     }
37     destroip(&A);
38     destroip(&B);
39     return 0;
40 }

```

Saída

```

1: morelli@morelli-archlinux:~/Documentos/Fatec/EstruturaDados/Pilha/Tarefa
.../Pilha/Tarefa > ./output/exercicio2
Digite 10 numeros aleatorios:
6
5
3
4
6
1
5
4
3
4
Vamos ordenar inversamente e garantir a unicidade...
Pilha ordenada inversamente (decrescente) e sem repetidos:
6 5 4 3 1

```

Tarefa 3

```

1  #include "pilha.h"
2  #include <stdio.h>
3  #include <string.h>
4
5  int main(void) {
6      char frase[256];
7      printf("Digite uma frase \n");
8      fgets(frase, sizeof(frase), stdin);
9      frase[strcspn(frase, "\n")] = '\0';
10
11     Pilha P = pilha(strlen(frase));
12
13     for (int i = 0; frase[i] != '\0'; i++) {
14         if (frase[i] != ' ') {

```

```

15     empilha((void *) (long) frase[i], P);
16 } else {
17     while (!vaziap(P))
18         printf("%c", (char) (long) desempilha(P));
19     printf(" ");
20 }
21 }
22 while (!vaziap(P))
23     printf("%c", (char) (long) desempilha(P));
24
25 printf("\n");
26 destroip(&P);
27 return 0;
28 }

```

Saída

```

1: morelli@morelli-archlinux:~/Documentos/Fatec/EstruturaDados/Pilha/Tarefa
.../Pilha/Tarefa > ./output/exercicio3
Digite uma frase
subi no onibus
ibus on subino

```

Tarefa 5

```

1  #include "pilha.h"
2  #include <stdio.h>
3
4  int main(void) {
5      Pilha P = pilha(5);
6      char s[11];
7
8      for (int i = 1; i <= 3; i++) {
9          printf("? ");
10         fgets(s, 11, stdin);
11         empilha(s, P);
12     }
13
14     while (!vaziap(P)) {
15         puts(desempilha(P));
16     }
17     return 0;
18 }

```

Saída

```

1: morelli@morelli-archlinux:~/Documentos/Fatec/EstruturaDados/Pilha/Tarefa
.../Pilha/Tarefa > ./output/exercicio5
? um
? dois
? tres
tres

tres

tres

```

- Isso ocorre por que a pilha armazena o endereço de memória, e não a cadeia de caracteres em si, por isso, quando inserimos um dois três, em vez de retornar de forma invertida cada um, o retorno é o ultimo valor que o endereço de memoria armazena

Tarefa 6

```

1  #include "pilha.h"
2  #include <stdio.h>
3  #include <string.h>
4
5  int main(void) {
6      Pilha P = pilha(5);
7      char s[11];
8
9      for (int i = 1; i <= 3; i++) {
10         printf("? ");
11         fgets(s, 11, stdin);
12         empilha(strdup(s), P);
13     }
14
15     while (!vaziap(P)) {
16         puts(desempilha(P));
17     }
18     return 0;
19 }

```

Saída

```
1: morelli@morelli-archlinux:~/Documentos/Fatec/EstruturaDados/Pilha/Tarefa
.../Pilha/Tarefa > ./output/exercicio6
? um
? dois
? tres
tres

dois

um
```